UNIVERSITÉ DE MONTRÉAL

MODELING AND CONTROL OF A FLEXIBLE SPACE ROBOT TO CAPTURE A TUMBLING DEBRIS

VINCENT DUBANCHET
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE ÉLECTRIQUE)
OCTOBRE 2016

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée:

MODELING AND CONTROL OF A FLEXIBLE SPACE ROBOT TO CAPTURE A TUMBLING DEBRIS

présentée par: DUBANCHET Vincent
en vue de l'obtention du diplôme de: Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de:

M. GOURDEAU Richard, Ph. D., président
M. SAUSSIÉ David, Ph. D., membre et directeur de recherche
Mme BÉRARD Caroline, Doctorat, membre et codirectrice de recherche
M. ALAZARD Daniel, H.D.R., membre et codirecteur de recherche
Mme AKHRIF Ouassima, Ph. D., membre
M. BENNANI Samir, Ph. D., membre

DEDICATION

To all my family, for its endless love and support...

ACKNOWLEDGEMENTS

This co-thesis was performed between the École Polytechnique of Montreal and the French engineering school ISAE-Supaéro at Toulouse, at the crossroads of two distinct educational system. Beyond the technical and teaching excellence of my teachers, their moral support and their constant help were crucial to succeed in such a project. I would like to thank as warmly and sincerely as possible David Saussié at Montreal, as well as Daniel Alazard and Caroline Bérard at Toulouse. In this long journey, you managed to guide me, to push me and to motivate me to always give the best of myself during all my researches and through my writing. It was a real chance to count you as my supervisors and if I had to start again, it would be with the same enthusiasm!

Nevertheless, the Ph.D. degree is not solely based on the work provided, but, above all, on the evaluation and endorsement of experts in the field. I am therefore extremely grateful to the external members of my jury: Richard Gourdeau, Ouassima Akhrif and Samir Bennani. I would like to insist that I was deeply honored that they accepted to report on my thesis. I thank them for taking the time and the care to evaluate my work, and for having provided so many ways of improvement and constructive comments for the future.

I would also like to thank Thales Alenia Space for co-funding this project and for allowing me to validate my results on their robotic test bench during six months. More precisely, I thank Catherine Le Peuvédic for her coaching throughout the thesis, as well as Adrien Grynagier, Xavier Roser and Carole Billot for their help during my internships in Cannes.

This thesis also received a valuable support from the France-Quebec scholarship *Frontenac*, to promote the mobility of Ph.D. students involved in a co-thesis. This grant allowed to fund one working stay per year in Montreal during the three years of the project, as well as the trip of David Saussié for the thesis defense in Toulouse.

Now, I want to come back to the people I met in the two schools, both for working purposes or during my many administrative issues.

In Montreal, I would like to express my sincere friendship to my former supervisor Lahcen Saydy for all his life advices, to Suzanne Le Bel for her culinary gifts, to Marc Charbonneau for his good mood, and to Nathalie Lévesque for her kindness and constant help. I cannot forget the work atmosphere of the laboratory, so inspiring thanks to all my colleagues from

all over the world: with Alexandre, Caroline, Antoine, Hugo, Lyes, Rabih, Meisam, Saad, Kai-Jun..., and with all the trainees including Florian, Benoit and Camille, to name but a few. In the same way, I had the chance to meet extremely warm colleagues during my internships at Thales with Carole, François and Francois, Juhaina, Hugues, Benoit, Quentin, Marie, Sabrina, Lorenzo and Francis. Finally, the end of my thesis in Toulouse was greatly enlivened and brightened by Régine Leconte, Stéphanie Lizy-Destrez and Bénédicte Escudier, as well as Françoise Loytier, Marie Fauré, Thierry Duigou, Alain Bouinot and Etienne Perrin. Similarly, I cannot forget my last "academic" colleagues with these technical or philosophical debates while playing some darts with Yann, Emilien and Fabien.

Before concluding, I would like to mention the closest friends I had during these last years, to tell them once again my deepest gratitude and sincere friendship for the good times and, above all, for the shoulder they offered me in the most difficult moments. A huge "Thank you" to Toto, Tilda, Laure, Phil and Ben in Montreal, to Emilien, Geoffrey, Bastien and Valentin in Toulouse, and finally to Angie to thank her for giving me so much love and motivation all along the road.

I finally dedicate this degree and these years of hard work to all my beloved family. Some people think that science is built "on the shoulders of giants", but I truly believe that I relied much more upon those of my family during this thesis... The love of my parents, of my brother, of my grandparents and of all my uncles and aunts had nurtured my efforts until the end and had given me the strength to make this great challenge successful.

Without the contribution of all these people, this step of my life would have never had the same flavor and I would not have thriven so much ... Thanks for everything!

RÉSUMÉ

La conquête spatiale des 60 dernières années a généré une grande quantité d'objets à la dérive sur les orbites terrestres. Leur nombre grandissant constitue un danger omniprésent pour l'exploitation des satellites, et requiert aujourd'hui une intervention humaine pour réduire les risques de collision. En effet, l'estimation de leur croissance sur un horizon de 200 ans, connue sous le nom de "syndrôme de Kessler", montre que l'accès à l'Espace sera grandement menacé si aucune mesure n'est prise pour endiguer cette prolifération. Le scientifique J.-C. Liou de la *National Aeronautics and Space Administration (NASA)* a montré que la tendance actuelle pourrait être stabilisée, voire inversée, si au moins cinq débris massifs étaient désorbités par an, tels que des satellites en fin de vie ou des étages supérieurs de lanceur. Parmi les nombreux concepts proposés pour cette mission, la robotique s'est imposée comme une des solutions les plus prometteuses grâce aux retours d'expérience des 30 dernières années. La *Station Spatiale Internationale (ISS)* possède déjà plusieurs bras robotiques opérationnels, et de nombreuses missions ont démontré le potentiel d'un tel système embarqué sur un satellite. Pour deux d'entre elles, des étapes fondamentales ont été validées pour le service en orbite, et s'avèrent être similaires aux problématiques de la désorbitation des débris.

Cette thèse se concentre sur l'étape de capture d'un débris en rotation par un bras robotique ayant des segments flexibles. Cette phase comprend la planification de trajectoire et le contrôle du robot spatial, afin de saisir le point cible du débris de la façon la plus délicate possible. La validation des technologies nécessaires à un tel projet est quasiment impossible sur Terre, et requiert des moyens démesurés pour effectuer des essais en orbite. Par conséquent, la modélisation et la simulation de systèmes multi-corps flexibles est traitée en détails, et constitue une forte contribution de la thèse. À l'aide de ces modèles, une validation mixte est proposée par des essais expérimentaux, en reproduisant la cinématique en orbite par des manipulateurs industriels contrôlés par une simulation en temps réel. En résumé, cette thèse est construite autour des trois domaines suivants : la modélisation des robots spatiaux, le design de lois de contrôle, et leur validation sur un cas test.

Dans un premier temps, la modélisation de robots spatiaux en condition d'apesanteur est développée pour une forme "en étoile". Cette hypothèse suppose que le système est composé d'une base rigide sur laquelle sont attachés des appendices flexible formés de corps en série. Les méthodes de modélisation de Lagrange et de Newton-Euler sont utilisées conjointement

pour représenter les flexibilités, et pour calculer la dynamique de façon efficace du point de vue numérique. Les robots rigides à base fixe sont rapidement décrits pour introduire les notations principales, puis les systèmes flexibles sont traités, pour enfin terminer par les robots spatiaux avec une base mobile. Tout au long de ces développements, la méthode du *Decoupled Natural Orthogonal Complement (DeNOC)* permet d'obtenir les modèles non-linéaires utilisés en simulation, ainsi que leur version linéarisée pour la synthèse et l'analyse des correcteurs.

Pour ce faire, la seconde partie met l'accent sur la génération de trajectoires de capture et sur le design global de la structure de contrôle. À la lumière de la théorie du contrôle optimal et du principe du minimum de Pontryagin, les trajectoires traditionnelles sont revisitées pour étendre la continuité au moment du contact aux termes d'accélération. Ainsi, le point cible est atteint délicatement, et son suivi avant le verrouillage du mécanisme de capture se fait sans à-coups au niveau des efforts de commande. Un correcteur global est ensuite proposé pour parcourir précisément ces trajectoires, en coordonnant le mouvement du satellite support avec celui de son bras robotique. La méthode de synthèse H_∞ structurée est utilisée pour obtenir des correcteurs d'ordre faible, suffisamment simples pour être embarqués sur les processeurs spatiaux actuels. Une analyse de robustesse est aussi menée sur la stabilité et les performances du système en boucle fermée, lorsque la configuration du bras change. À partir du formalisme de la *Linear Fractional Transformation (LFT)*, un modèle minimal est d'abord obtenu pour représenter une matrice de rotation en fonction de son angle, permettant ensuite de paramétrer la dynamique globale par l'ensemble des angles aux articulations. Une μ-analyse est menée sur ces modèles pour extraire les bornes maximales de variation des angles garantissant le stabilité et la performance robuste en boucle fermée.

Enfin, la validation de la structure de contrôle est obtenue à deux niveaux : d'abord numériquement avec un simulateur haute-fidélité, et ensuite expérimentalement avec un banc d'essais robotique. Le schéma de simulation reproduit le comportement du robot spatial en apesanteur ainsi que celui du débris en rotation. Les non-linéarités dues aux segments flexibles sont intégralement prises en compte dans le calcul de la dynamique. La capture du débris est alors assurée en maintenant l'erreur de suivi de la trajectoire optimale sous une tolérance donnée, et ce, tout au long de la simulation. Au niveau expérimental, le simulateur est embarqué sur un module temps-réel afin de contrôler les deux robots industriels. Ils reproduisent simultanément la cinématique en apesanteur de l'effecteur du robot spatial, d'une part, et du point cible du débris, d'autre part. Une caméra et son algorithme de traitement d'images sont incorporés dans la boucle de mesure du correcteur, de manière à réaliser des validations matérielles, aussi qualifiées de *Hardware-In-the-Loop (HIL)*.

Pour résumer, cette thèse contribue dans un premier temps à la modélisation de robots spatiaux ayant un nombre arbitraire d'appendices flexibles, dans les deux buts distincts de leur simulation et de leur contrôle. De plus, la génération des trajectoires a été adaptée pour obtenir la continuité en position, en vitesse et en accélération au moment de la capture, et un correcteur robuste est synthétisé et analysé pour assurer un suivi précis de ces dernières. Finalement, les validations expérimentales en boucle fermée constituent la dernière contribution majeure de cette thèse, avec l'utilisation d'éléments physiques en temps réel au sein de simulations numériques.

ABSTRACT

After 60 years of intensive satellite launches, the number of drifting objects in Earth orbits is reaching a shifting point, where human intervention is becoming necessary to reduce the threat of collision. Indeed, a 200 year forecast, known as the "Kessler syndrome", states that space access will be greatly compromised if nothing is done to address the proliferation of these debris. Scientist J.-C. Liou from the *National Aeronautics and Space Administration (NASA)* has shown that the current trend could be reversed if at least five massive objects, such as dead satellites or rocket upper stages, were de-orbited each year. Among the various technical concepts considered for debris removal, robotics has emerged, over the last 30 years, as one of the most promising solutions. The *International Space Station (ISS)* already possesses fully operational robotic arms, and other missions have explored the potential of a manipulator embedded onto a satellite. During two of the latter, key capabilities have been demonstrated for on-orbit servicing, and prove to be equally useful for the purpose of debris removal.

This thesis focuses on the close range capture of a tumbling debris by a robotic arm with light-weight flexible segments. This phase includes the motion planning and the control of a space robot, in order to smoothly catch a target point on the debris. The validation of such technologies is almost impossible on Earth and leads to prohibitive costs when performed on orbit. Therefore, the modeling and simulation of flexible multi-body systems has been investigated thoroughly, and is likewise a strong contribution of the thesis. Based on these models, an experimental validation is proposed by reproducing the on-orbit kinematics on a test bench made up of two industrial manipulators and driven by a real-time dynamic simulation. In a nutshell, the thesis is built around three main parts: the modeling of a space robot, the design of control laws, and their validation on a test case.

The first part is dedicated to the flexible modeling of a space robot in conditions of weightlessness. A "star-shaped" multi-body system is considered, meaning that the rigid base carries various flexible appendages and robotic arms, assumed to be open mechanical chains only. The classic Newton-Euler and Lagrangian algorithms are brought together to account for the flexibility and to compute the dynamics in a numerically efficient way. The modeling step starts with the rigid fixed-base manipulators in order to introduce the notations, then, details the flexible ones, and ends with the moving-base system to represent the space robots. Using the *Decoupled Natural Orthogonal Complement (DeNOC)* approach, a complete nonlinear

model is developed for simulation, while a linearized version is extracted for the synthesis and the analysis of the controllers.

To this end, the second part focuses on the path planning and the design of a global control law. Classic capture trajectories are revisited in the scope of optimal control theory with the Pontryagin's minimum principle, in order to reach the acceleration continuity at the instant of grasping. By ensuring this latter in addition to the position and the speed continuity, the target point can be reached and then tracked before grasping, without any discontinuity in the control efforts. A coordinated controller is then introduced to manage both the spacecraft and the arm motions, in order to accurately follow these trajectories. The fixed-structure H_∞ framework is used to design reduced-order controllers, which stay simple enough to be compatible with current space processors capabilities. A robustness analysis is also performed to assess their stability and their performances when the joint angles are varying. Based on the *Linear Fractional Transformation (LFT)* formalism, a minimal model is first derived for a rotation matrix as a function of its angle, and then, it is used recursively to parameterize the whole dynamics by the set of joint angles defining the arm configuration. Several μ-analyses are performed to obtain the maximum joints motion before losing the stability and the performances of the closed-loop system.

Finally, the validation is carried out by a twofold approach: numerically with a high-fidelity simulator, and experimentally with a robotic test bench. The simulation scheme reproduces the behavior of the space robot and of the tumbling debris in weightlessness. It includes the whole nonlinearities coming from the flexible segments dynamics. The debris capture is ensured by maintaining, throughout the simulation, the trajectory tracking error below a given tolerance. Then, this simulator is embedded on a real-time module to control two industrial manipulators that reproduce the kinematics of, respectively, the end-effector of the space robot and the target point of the debris. A camera and its image processing algorithm are introduced in the loop to provide the controller feedback, and thus, allow to perform a *Hardware-In-the-Loop (HIL)* validation.

As a conclusion, this thesis first contributes to the modeling of space robots with an arbitrary number of flexible appendages, for the purposes of simulation and control. In addition, a path planning algorithm is derived to ensure the position, the speed, and the acceleration continuity at the instant of capture, and a robust controller is synthesized and analyzed to efficiently track this desired trajectory. Eventually, the closed-loop experiments are the last main contribution of the thesis, with the real-time interfacing of physical components with numerical simulations.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ABA *Articulated-Body Algorithm.*

ACS *Attitude Control System.*

ADR *Active Debris Removal.*

AMM *Assumed Mode Method.*

AOCS *Attitude and Orbital Control System.*

ASAT *Anti-SATellite weapon.*

ASTRO *Autonomous Space Transport Robotic Orbiter.*

ATV *Automated Transfer Vehicle.*

CMG *Control Moment Gyroscope.*

CNES *Centre National d'Études Spatiales.*

CoM *Center of Mass.*

COPUOS *Committee On the Peaceful Uses of Outer Space.*

CRBA *Composite Rigid Body Algorithm.*

CSA *Canadian Space Agency.*

DARPA *Defense Advanced Research Projects Agency.*

DeNOC *Decoupled Natural Orthogonal Complement.*

DEOS *DEutsche Orbitale Servicing mission.*

DH *Denavit Hartenberg.*

DLR *Deutsches zentrum für Luft- und Raumfahrt.*

DoF *Degree-of-Freedom.*

ECI *Earth-Centered Inertial.*

EKF *Extended Kalman Filter.*

EPOS *European Proximity Operations Simulator.*

ERA *European Robotic Arm.*

ESA *European Space Agency.*

ETS *Engineering Test Satellite.*

EVA *Extra-Vehicular Activity.*

FEM *Finite Element Method.*

FREND *Front-end Robotics Enabling Near-term Demonstration.*

GJM *Generalized Jacobian Matrix.*

GPS *Global Positioning System.*

HIL *Hardware-In-the-Loop.*

HTV *H-II Transfer Vehicle.*

IADC *Inter-Agency space Debris Coordination Committee.*

IMU *Inertial Measurement Unit.*

ISS *International Space Station.*

JAXA *Japan Aerospace eXploration Agency.*

JEMRMS *Japanese Experiment Module Remote Manipulator System.*

KF *Kalman Filter.*

LEO *Low Earth Orbit.*

LFT *Linear Fractional Transformation.*

LHS *Left Hand Side.*

LPV *Linear Parameter Varying.*

LQR *Linear Quadratic Regulator.*

MDA *MacDonald, Dettwiler and Associates.*

MIT *Massachusetts Institute of Technology.*

NASA *National Aeronautics and Space Administration.*

NASDA *NAtional Space Development Agency.*

NBV *Neutral Buoyancy Vehicle.*

NextSat *Next generation Satellite.*

NOC *Natural Orthogonal Complement.*

NORAD *North American Aerospace Defense Command.*

ODE *Ordinary Differential Equation.*

OE *Orbital Express.*

OLEV *Orbital Life Extension Vehicle.*

ORU *Orbital Replacement Unit.*

OST *Outer Space Treaty.*

PD *Proportional-Derivative.*

PDE *Partial Differential Equation.*

PID *Proportional-Integral-Derivative.*

RGE *Recursive Gaussian Elimination.*

RHS *Right Hand Side.*

RMS *Root Mean Square.*

RNEA *Recursive Newton-Euler Algorithm.*

RNS *Reaction Null-Space.*

ROKVISS *RObot Komponent Verification on ISS.*

RSSS *Russian Space Surveillance System.*

RW *Reaction Wheel.*

SDT *Satellite Dynamics Toolbox.*

SRMS *Shuttle Remote Manipulator System.*

SSN *Space Surveillance Network.*

SSTS *Space Surveillance and Tracking Segment.*

SUMO *Spacecraft for the Unmanned Modification of Orbits.*

TAS *Thales Alenia Space.*

TCS *Thermal Control System.*

TECSAS *TEChnology SAtellites for demonstration and verification of Space systems.*

TLE *Two-Line Elements.*

TRL *Technology Readiness Level.*

UN *United Nations.*

US *United States.*

LIST OF APPENDICES

LIST OF PUBLICATIONS

The main contributions of the thesis have led to various publications in conferences and journals. The most recent ones are still being finalized, and will soon be submitted to the corresponding journal.

(Dubanchet et al., 2014) V. Dubanchet, D. Saussié, D. Alazard, C. Bérard, and C. Le Peuvédic, "Modeling and control of a space robot for active debris removal", in *Proceedings of the 9th International ESA Conference on Guidance, Navigation and Control*, 2014, volume 1, pages 1–19.

(Dubanchet et al., 2015a) V. Dubanchet, D. Saussié, D. Alazard, C. Bérard, and C. Le Peuvédic, "Modeling and control of a space robot for active debris removal", in *CEAS Space Journal*, volume 7, issue 2, pages 203–218, 2015.

(Dubanchet et al., 2015b) V. Dubanchet, D. Saussié, D. Alazard, C. Bérard, and C. Le Peuvédic, "Motion Planning and Control of a Space Robot to Capture a Tumbling Debris", in *Advances in Aerospace Guidance, Navigation and Control: Selected Papers of the Third CEAS Specialist Conference*, Springer Publishing, volume 1, pages 699–717, 2015.

(Dubanchet et al., 2016a) V. Dubanchet, D. Saussié, D. Alazard, C. Bérard, and C. Le Peuvédic, "Influence of corrective terms in flexible manipulators dynamics using the DeNOC approach", in (to be submitted to) *Journal of Dynamic Systems, Meas., and Control (ASME)*, 2016.

(Dubanchet et al., 2016b) V. Dubanchet, D. Saussié, D. Alazard, C. Bérard, and C. Le Peuvédic, "LFT Modeling and Robust Performances Analysis of a Rigid Multi-Body System", in (to be submitted to) *IEEE Transactions on Robotics*, 2016.

CHAPTER 1 INTRODUCTION

Space science and technology have known a tremendous leap forward in the last century, bringing into orbit all kinds of satellites and rocket stages. Nowadays, most of these objects are no longer functional and are classified as *space debris*. They have multiplied at an increasing pace over the last years, reaching a tipping point where their natural decay with the atmospheric drag is no longer sufficient to counter this rise. In 1978, the scientists D. J. Kessler and B. G. Cour-Palais pointed out for the first time that such a density of orbiting bodies would lead to an increasing probability of collision, that could create a debris belt around the Earth (Kessler and Cour-Palais, 1978) (see Figure 1.1). Exploring these probabilistic scenarios through extensive simulations, J.-C. Liou eventually demonstrated that the "Kessler syndrome" is already engaged, meaning that, the debris would multiply in an unstoppable chain reaction without human intervention (Liou and Johnson, 2009). The only remaining uncertainty in these studies is whether the loss of space access will happen in a near or far future, depending on the growth rate of the debris population.

These objects are giving rise to two main issues: they threaten the functional satellites and endanger the astronauts aboard the *International Space Station (ISS)* or during *Extra-Vehicular Activity (EVA)* (Selding, 2014; Woods, 2015). Due to their very high speed, around 10 km/s, they can cause serious damages depending on their size. In addition, they will stay in orbit for years or centuries depending on their trajectory. According to the French space agency *Centre National d'Études Spatiales (CNES)*, the actual debris population is divided into three main categories:

- 350,000,000 debris smaller than 1 cm, of **low risk** ;
- 300,000 debris between 1 cm and 10 cm, of **high risk** ;
- 16,000 debris bigger than 10 cm, of **moderate risk**.

Among these, the medium size debris are the most hazardous because they are too small to be tracked by ground systems, but big enough to cause disastrous damages on a functional satellite (Johnson, 2009). Actual materials for shielding allow the satellite to endure impacts with the smallest ones, and the biggest ones are tracked from ground to prevent any impact by performing avoidance maneuvers.

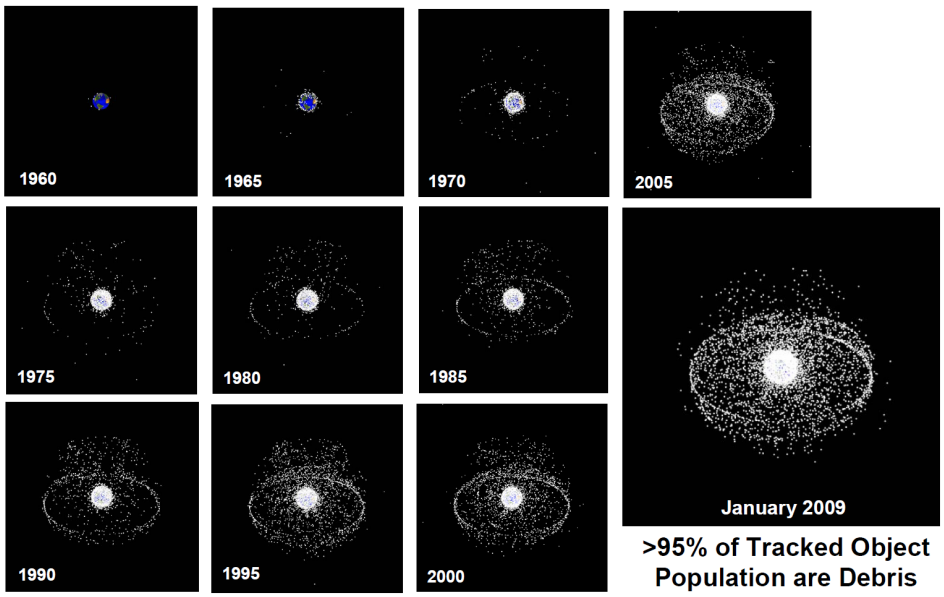


Figure 1.1: Evolution of the debris population over the space exploration.
Source: NASA - http://orbitaldebris.jsc.nasa.gov/photogallery/beehives.html#geo

1.1 Space Debris Mitigation

Following the recommendations of J.-C. Liou in (Liou et al., 2010), at least 5 massive debris must be deorbited per year to stabilize the actual population and avoid the Kessler syndrome. Indeed, these objects are more likely to create new debris by colliding with one another and hence represent the main objective of mitigation missions. In the next section, a general and technical overview is given over the main issues involved in a debris removal mission, from the legal framework to technical considerations.

1.1.1 Today's issues

The space debris issue is mainly tackled by a working group of the *United Nations (UN)* called the *Committee On the Peaceful Uses of Outer Space (COPUOS)*. It provides many data about the actual space pollution, and draws some recommendations and guidelines to reduce as much as possible the current human footprint (COPUOS, 1999). Among its members, the most influent space agencies around the world also created the *Inter-Agency space Debris Coordination Committee (IADC)* in 1993, which aims at cooperating more closely to perform the necessary technical studies.

Recommendations For example in 2007, the IADC wrote the *Space Debris Mitigation Guidelines*, and this document served as a reference for the COPUOS report (COPUOS, 2010). These recommendations encourage end-of-life procedures to deorbit satellites and

rockets, and aim at preventing catastrophic collisions or breakups. Indeed, two main events worsened the situation in the 2000s. In 2007, the Chinese decided to test an Anti-Satellite missile in *Low Earth Orbit (LEO)* on their *Fengyun-1C* satellite, and in 2009, a collision occurred between an American satellite, *Iridium 33*, and a Russian one, *Cosmos 2251*. These events increased the number of debris of 4,000 and 2,500 pieces, respectively.

Surveillance Most of the actual debris greater than 10 cm can be tracked from the ground thanks to surveillance networks like the *United States (US)* facility called *Space Surveillance Network (SSN)*, capable of detecting objects greater than 10 cm (Wright, 2010) in LEO. The measurements are turned into the famous *Two-Line Elements (TLE)* dataset by the *North American Aerospace Defense Command (NORAD)*, in order to describe the orbital parameters of these objects. These data are available for example https://celestrak.com. Thanks to such a system, the evolution of space debris population has been observed since the launch of the first satellite, *Sputnik*, by the Russians on October 4^{th}, 1957. The result is illustrated in Figure 1.2.

For autonomy purposes, the Russians also own a similar space surveillance system, the *Russian Space Surveillance System (RSSS)* also known as SKKP. Even though currently less efficient, the *European Space Agency (ESA)* is developing a third major facility called *Space Surveillance and Tracking Segment (SSTS)*.

Selection Once the space debris are cataloged, one needs to choose the most relevant objects to deorbit. For example, in (Couzin et al., 2013), a mission-based approach is used, so space debris are sorted for a given range of orbital inclination and altitude. It highlights which groups are reachable with a minimum of maneuvers in a multi-target removal scenario. On the other hand, from a risk-based approach, C. Bonnal focuses on the criticality of each object in orbit. Its size and mass strongly influence its probability of collision with others, as well as the overcrowding of its orbit. In the same way, the top 500 objects with the highest collision probability are drawn up in (Liou, 2011), and provide a solid starting point.

Legal Framework Prior to the treatment of any debris, one major issue remains about the definition of its *nationality*. This is supposed to determine the liability of a given country to clean it up. As emphasized by (Weeden, 2011; Ansdell, 2010), many definitions can be adopted to choose this country: is it the one that built the object? that launched it? that operated it? All these questions are still open, but steps forward have been made with the *United Nations (UN)*, through the *Outer Space Treaty (OST)* defined in 1967, and the Liability Convention adopted in 1972. This legal framework is also supposed to dictate how the price of an active debris removal mission must be shared between the liable countries.

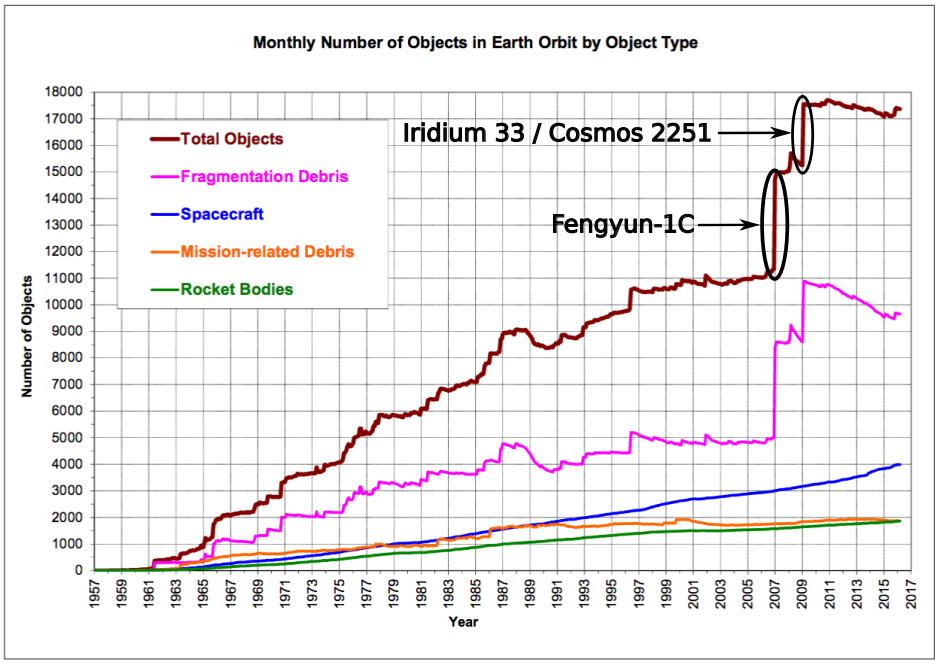


Figure 1.2: Monthly number of cataloged objects in Earth orbit, tracked by the *Space Surveillance Network (SSN)*. The last two peaks are due to the breakup of the Chinese satellite *Fenguyun-1C* in 2007, and to the collision of Iridium/Cosmos in 2009.
Source: *National Aeronautics and Space Administration (NASA)* - Orbital Debris Quarterly News - April 2016 - vol. 20 (1-2), annotated.

Disintegration Eventually, the debris disintegration is the last source of concern. The consumed portion of a body during the atmospheric re-entry is hardly known due to its unpredictable nature, which fostered active researches on the matter. For example, a complete simulation software called SCARAB is proposed in (Fritsche et al., 2000). All the components of a satellite can be specified in terms of material and shapes, and simulations are performed at structural and thermal levels to predict the level of disintegration on a given re-entry orbit. This issue drives C. Bonnal to recommend, in (Bonnal et al., 2013), an *Active Debris Removal (ADR)* scenario with a controlled target trajectory, instead of a simple uncontrolled re-entry.

1.1.2 Potential technical solutions

The emphasis is now put on the technical solutions that allows to safely deorbit a space debris. Some of them provide remote effects or direct efforts in the common goal to lower the debris altitude and drive it to a re-entry trajectory that will consume it entirely. A major difference must also be made between the deorbitation concepts and the capture devices requiring a

physical contact. In the sequel, the term "non-cooperative" means that the capture must be performed for tumbling objects without any visual marker nor grappling fixtures (Bonnal et al., 2013).

Deorbitation devices Among the many solutions proposed in the literature, they first differ by the way they act on the debris. The choice is made here to present the ground-based concepts and the on-orbit ones as two different families.

• **From ground:** As mentioned earlier with the Chinese satellite breakup, an *Anti-SATellite weapon (ASAT)* missile could be launched directly from the ground to intercept and destroy a space debris. This idea is obviously left aside for ADR missions since it only dismantle a big debris into smaller pieces and does not solve their threat.
Another remote action is possible through the laser ablation technology presented by (Phipps et al., 1996). It consists in hitting the debris with a high-energy pulsed laser in order to locally ablate material, as shown in Figure 1.3. The resulting jet of plasma creates an impulse force on the debris by the reaction principle. This approach remains only effective in a short amount of time for the smallest debris (<10 cm), but can also be considered to slightly lower the altitude of massive debris in order to reduce the threat they represent on crowded orbits.

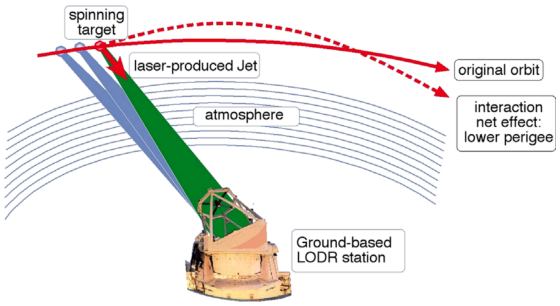
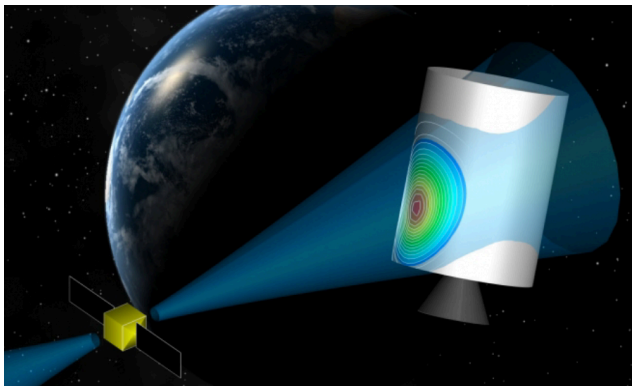


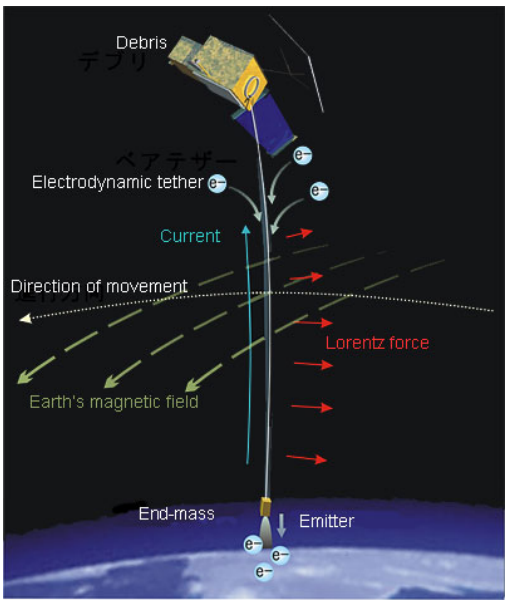
Figure 1.3: Laser Orbital Debris Removal (LODR) concept by C. Phipps.
Source: (Phipps, 2014)

• **On-orbit:** Due to the hazardous re-entry of a debris treated from the ground, most of actual concepts turn to on-orbit scenarios. Again, both kind of devices are encountered to exert a force on the debris, with or without contact.

– **Contactless:** As a variant of the previous idea, space-based lasers are proposed in (Schall, 1998) and (Phipps, 2014). The main advantages are to avoid the crossing of the atmosphere and to allow for the use of lasers of lower energy. Instead of creating the plasma by ablating matter, M. Merino *et al.* (Merino et al., 2011) proposed to generate it on board, and then to throw it toward the debris to drive it, as shown in Figure 1.4a. In the same way, an ESA study proposes to form a foam ball around the debris by throwing it from the chaser (Andrenucci et al., 2011). It would increase the



(a) Source: (Wormnes et al., 2013)



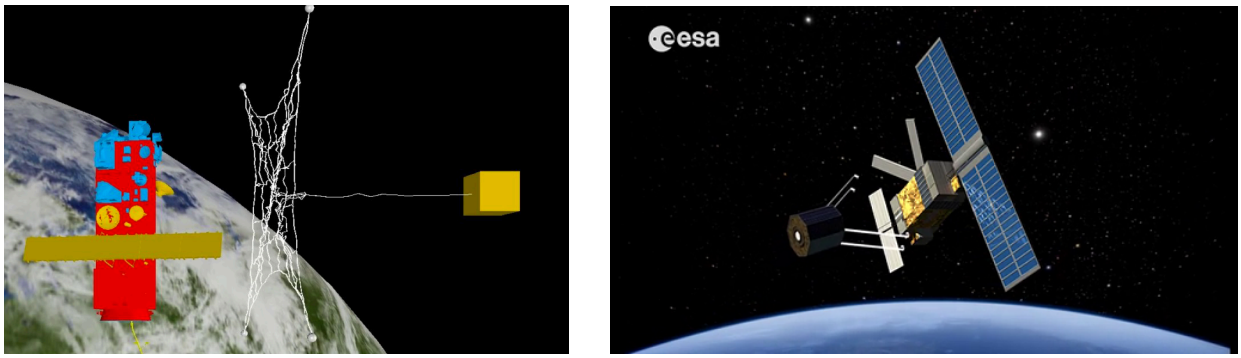
(b) Source: JAXA

Figure 1.4: Examples of on-orbit deorbitation methods with and without contact. (a) Ion beam thrown toward a debris; (b) Concept of an electrodynamic tether.

surface to mass ratio of the debris, and consequently improve the drag effect to lower its altitude.

Other contactless concepts focus on magnet-based actuators embedded on a chaser to move the debris with the induced Eddy currents (Ortiz Gómez et al., 2016). In addition to the small forces applied, this approach requires a very accurate formation flight between the chaser and the target (Fabacher et al., 2015).

– **Contact:** Concerning the devices which need a physical contact with the debris, one can find electrodynamic tethers, solar sails or even deorbiting kits with a small engine (Bonnal et al., 2013). The first ones are long wires suspended to the object, and undergo a drag force due to the Earth magnetic field, which is transmitted to the debris, as illustrated in Figure 1.4b. A thorough overview to assess the pros and cons is given in (Pardini et al., 2009), where the probability of impact with small orbiting debris is evaluated for these appendages spreading over many kilometers. To give an idea, in 1996, the French satellite *Cerise* lost its mast of only 3 m by a collision with an orbiting debris (Alby et al., 2007). Eventually, the last option with a deorbiting kit would be the best solution for a controlled re-entry (Couzin et al., 2013). Nevertheless, the kit itself is much heavier than the tether solution, since it contains a rocket with its propulsion and can reach hundreds of kilograms.



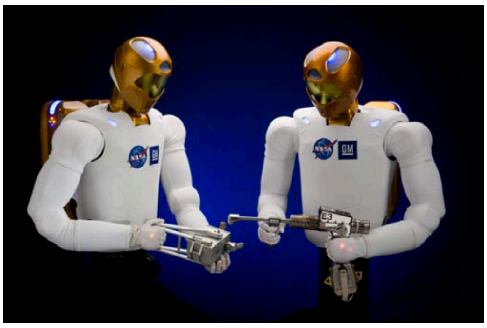
(a) Source: (Wormnes et al., 2013) (b) Source: ESA - eDeorbit

Figure 1.5: Most promising methods to capture a debris.
(a) Pulling technology with a net grasping; (b) Pushing technology with a clamping mechanism.

As a conclusion about the deorbitation technologies, some of them appear more suited for small debris, like the laser ablation or the foam expansion, and others are more effective for big debris, like the electrodynamic tethers or the deorbiting kits. For the latter, a capture is necessary to append the kit on the debris.

Capture devices Again, many solutions are under development, like nets, harpoons or robotic arms. A survey is provided in (Wormnes et al., 2013) from the ESA point of view. Pulling technologies using nets and harpoons are emphasized for mono-target scenario. The chaser would capture only one debris and then pull it toward the Earth to burn with it (see Figure 1.5a). In a multi-target scenario, solid propulsion deorbiting kits offer a better compromise since a chaser could embed many of them in a single mission (Castronuovo, 2011). Focusing on this concept, C. Bonnal comes to the conclusion that a robotic arm is the best compromise to perform both capture and kit fastening (Bonnal et al., 2013). In addition, this technology is the main option for on-orbit servicing and its *Technology Readiness Level (TRL)* is already higher than most of its competitors (NASA, 2010) (see Figure 1.5b).

According to the previous analyses, space robotics emerges as one of the key technology for active debris removal, as well as the nets and harpoons concepts. The present thesis naturally builds upon the strengths of the robotic arm solution. In the sequel, a typical mission is described for such a capture mechanism. Within this description, the parts addressed by this thesis are highlighted. A review of the past and current robotic missions for both on-orbit servicing and debris capture is also provided to enhance the remaining fields of improvement for these technologies.



(a) Source: (Diftler et al., 2011)

(b) Source: (Fuchs et al., 2009)

Figure 1.6: Most advanced project of humanoid robotics in space.
(a) Robonaut2 project from NASA and General Electric ; (b) Rollin'Justin from the DLR.

1.2 On-Orbit Space Robotics

The latest developments about all the fields of space robotics can be found in the surveys of K. Yoshida in (Yoshida, 2009), of L. Pedersen *et al.* in (Pedersen et al., 2003), or in the more technical paper of A. Flores-Abad *et al.* in (Flores-Abad et al., 2014). Apart from the in-space technologies, they also go deeply into mobile robotics, covering the vast fields of robotic motion on-ground, perception, image analysis, path planning in an a priori unknown environment, etc. Impressive humanoid robots are also being developed to help and replace astronauts in hazardous operating conditions. The more advanced ones are the American *Robonaut2*, which was already sent on the ISS (Diftler et al., 2011), and the German *Rollin'Justin* (Fuchs et al., 2009). They are respectively illustrated in Figure 1.6a and Figure 1.6b.

In the present work, the stress is put on the projects including robotic arms embedded on a spacecraft to capture cooperative or non-cooperative objects. This last point split the fields of application into two main branches: the on-orbit servicing and the active removal of massive debris. They both ask for similar technologies, but the main difference lies in the nature of the object to catch: for the debris removal issue, no handle or visual marker are present during capture, and the object can be rotating at high rates. The capabilities of the involved technologies have already been demonstrated on servicing scenarios, but remains an open question in the case of the on-orbit capture of a tumbling debris.

Many other missions followed the pioneering era of the *Shuttle Remote Manipulator Sys-*

tem (SRMS), the Canadian robotic arm embedded on the Space Shuttle in the 80's and nicknamed *Canadarm*. To name but a few, the Canadarm allowed to capture the famous Hubble telescope for repairs in the early 90's, the Japanese space agency *Japan Aerospace eXploration Agency (JAXA)* demonstrated a collaborative capture with the *Engineering Test Satellite (ETS)*-VII project in 1997, and the Americans performed autonomous rendezvous and semi-autonomous capture and berthing with the *Orbital Express (OE)* mission in 2006. They are presented into more details through the next sections to draw a brief overview of the maturity of actual technologies.

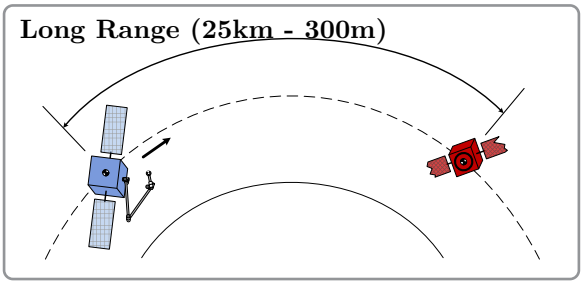
1.2.1 Typical mission scenario

The sequence of events usually performed for an autonomous rendezvous and capture is well described in (NASA, 2010) for servicing, and in (Rekleitis et al., 2007; Bonnal et al., 2013) for debris removal. To summarize, the mission has the following steps:

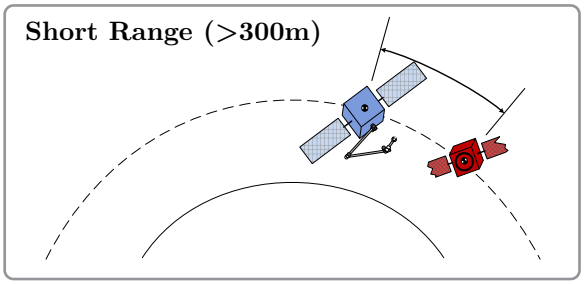
1. Long-Range Rendezvous: from 25 km to 300 m from the target (see Figure 1.7a);
2. Short-Range Rendezvous: from 300 m to 50 m (see Figure 1.7b);
3. Fly Around & Data acquisition: estimation and propagation of the target dynamics[1], path planning for the capture (see Figure 1.7c);
4. Capture: deployment of the capture mechanism, interception trajectory, and mechanical interfacing with the target; (see Figure 1.7d);
5. Post-Capture: de-tumbling and securing of the composite system (chaser+target), berthing to release the load on the capture mechanism (see Figure 1.7e);
6. Service Operations: refueling or unit change for servicing, or fixation of a deorbitation kit for ADR (depending on the strategy) (see Figure 1.7f);
7. Final Step: release for servicing, or deorbitation for ADR.

More details about the mission profile are available through (Castronuovo, 2011; Guariniello et al., 2011) from the debris removal point of view. One key point, which is still undergoing intensive tests, is the choice of the point of capture on the debris. Since the target does not possess any handle, the focus is set on common features, like the launcher ring or on the nozzle of the apogee kick motor. The first one is common to any satellite and its counterpart is also present on the orbiting rocket upper stages, making it more attractive to capture these

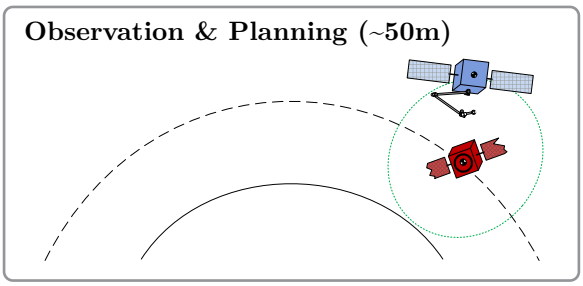
[1] See (Persson, 2015) for an innovative approach based on Kalman filtering



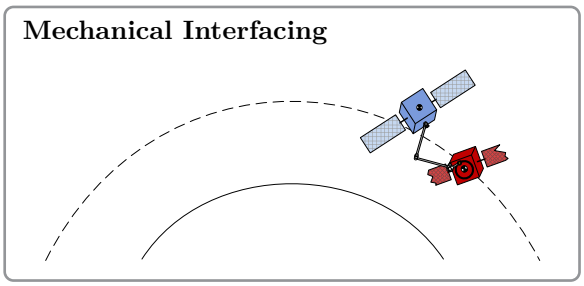
(a) Long-Range Rendezvous



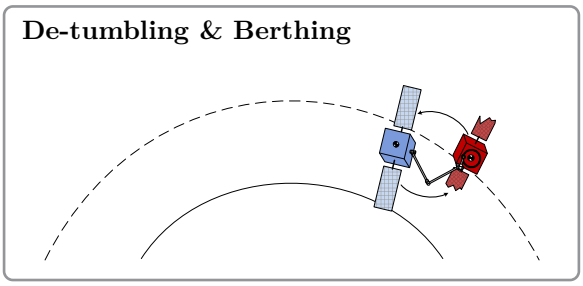
(b) Short-Range Rendezvous



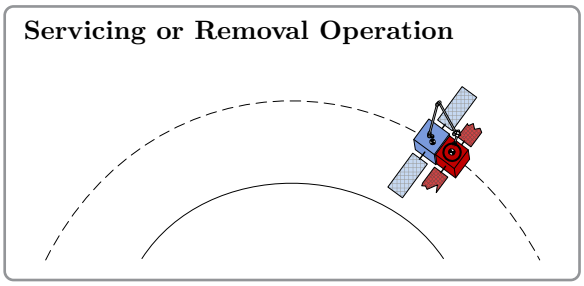
(c) Fly Around



(d) Capture



(e) Post-Capture



(f) Service Operations

Figure 1.7: Typical servicing/debris removal mission profile

two types of debris with the same system (Castronuovo, 2011; Obermark et al., 2007). The second one is most common for geostationary satellites, since they need an extra impulse to reach their final orbit. The nozzle provides a feature strongly enough to dock on it and then manipulate the whole spacecraft through it (Yoshida and Nakanishi, 2003; Boge et al., 2010).

1.2.2 Past missions overview

Among the few missions mentioned above, the three main ones involved a robotic capture with different levels of autonomy. A full description of each mission achievements regarding this autonomy and the task performed by the robotic arms is done in (Rekleitis et al., 2007).

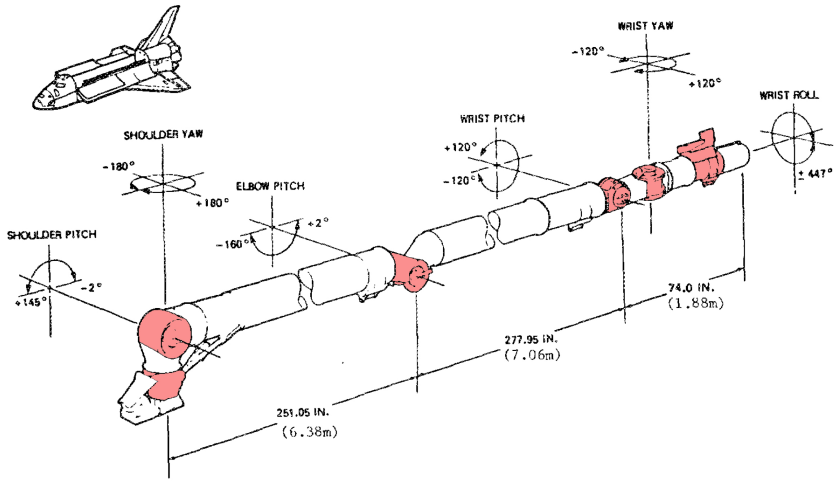


Figure 1.8: Human-like structure of the Canadarm (6 joints). Source: (Aikenhead et al., 1983)

The next paragraphs illustrate especially the robotic tasks performed during these missions, and do not extend to other interesting aspects, like the autonomous rendezvous. Indeed, this issue has already been demonstrated and is common for projects like the *Automated Transfer Vehicle (ATV)* by ESA (Wikipedia, 2013a), or the *H-II Transfer Vehicle (HTV)* by JAXA (Wikipedia, 2013b). Conversely, space robotics are still under advanced developments to increase their autonomy level. The Canadian robotic arm embedded on the Space Shuttle is presented first, then the contributions of the Japanese mission ETS-VII and of the American one Orbital Express are emphasized. For each of them, illustration are provided in Figure 1.9.

1.2.2.1 Canadian *SRMS*

Manually controlled robotic arm.

This arm was designed by a Canadian company called *SPAR Aerospace* at that time, and whose robotic division is now part of the *MacDonald, Dettwiler and Associates (MDA)* group. The arm was 15 meters long and had a structure similar to a human arm with a shoulder, a forearm, an elbow, an arm and a wrist, as illustrated in Figure 1.8. The two main beams were, respectively, around 6 and 7 meters long for a total mass of 450 kg (Aikenhead et al., 1983). Flexibility was a strong issue to control this arm because the first flexible mode frequencies were very low.

Reviews of its main missions are done in (Sachdev, 1986; Hiltz et al., 2001). Among the most important ones, the Canadarm used to inspect the thermal protection system of the Space

Shuttle before the atmospheric re-entry. In some cases, it reveals critical for the mission success and for the crew safety. For instance, it allowed to clean a venting port blocked with ice on the 41-D mission in 1984, and it regularly offered a safe base to maintain the astronaut during EVA.

In a nutshell, the Canadarm was the first space robotic arm and proved to be essential to carry out many tasks, from the repairing of malfunctioning satellites, to the inspection of the Space Shuttle itself, and even for the assembly of multi-pieces structures. But this arm was **controlled by the astronauts on-orbit**, and did not offer any autonomy.

1.2.2.2 Japanese *ETS-VII*

Semi-autonomous robotic capture of a cooperative target.

In the late 90's, the Japanese space agency, called at that time the *NAtional Space Development Agency (NASDA)*, performed a visionary mission with the ETS-VII project. Based on two satellites, a chaser *Hikoboshi* of 2.5 tons and a target *Orihime*[2] of 0.4 t, autonomous rendezvous and docking were carried out, including also robotic captures (Kasai et al., 1999). During the operations, the ground teleoperated the arm with a time delay of 6 to 7 seconds, thanks to relay satellites in geostationary orbit.

At the robotic level, the *Toshiba Co.* company designed its 6 *Degree-of-Freedom (DoF)* manipulator, with a mass of 45 kg, a span of 2 m and a power consumption of 80 W. It demonstrated a robotic release of the target with a relative motion of 1 mm/s in translation and 0.01°/s in rotation (Inaba and Oda, 2000). Thanks to the onboard image processing and the handle marker on the target, the robotic arm was remotely controlled to re-capture it from the ground. A coordinated control was designed to perform the required robotic motion, while compensating for the large disturbances induced on the spacecraft (Oda, 1996, 1999). A simplified feedforward compensation was used by approximating the angular momentum transfer from the arm to the base (Oda, 1994).

It is worth reminding that this experiment was the first in history to perform an autonomous rendezvous in short-range, a refueling and an *Orbital Replacement Unit (ORU)* transfer in semi-autonomous mode. The robotic arm was **manipulated from the ground** with semi-autonomous tasks, like "Move from point A to point B", or fully manually by driving its end-effector with joysticks (Kasai et al., 1999).

[2]From the names of a prince and a princess in a Japanese legend (Yoshida, 2009).

1.2.2.3 American *Orbital Express*

Autonomous capture of a cooperative target.

The last servicing demonstration took place with the *Orbital Express (OE)* mission sponsored by the American *Defense Advanced Research Projects Agency (DARPA)*. It consisted of two satellites, the chaser of 1.1 t called *Autonomous Space Transport Robotic Orbiter (ASTRO)*, and the target of 0.2 t named *Next generation Satellite (NextSat)*. The first one was equipped with a 6 DoF robotic arm designed by MDA, with a mass of 71 kg, a span of 3.3 m and a power consumption of 131 W. It was able to capture and service the target, to perform a refueling and a battery and ORU replacements (Friend, 2008).

The mission demonstrated many autonomous rendezvous and docking, even for far-range scenarios (>5 km), and performed autonomous captures with the robotic arm. The docking mechanism was a brand new concept developed in (Timmons and Ringelberg, 2008; Stamm and Motaghedi, 2004).

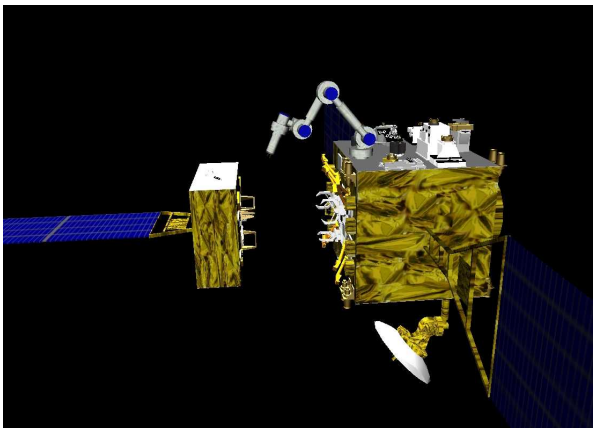
The robotic arm was once again mainly designed to perform the servicing tasks, and in few scenarios to capture and lead the target for a safe berthing. This was the first time that a **fully autonomous capture by a robotic arm** took place in space. But the object was a cooperative spacecraft with visual markers and a fixed attitude.

The next section presents the robotic arms operating in space, and the missions under development which aim at demonstrating a non-cooperative capture.

1.2.3 Current and future missions

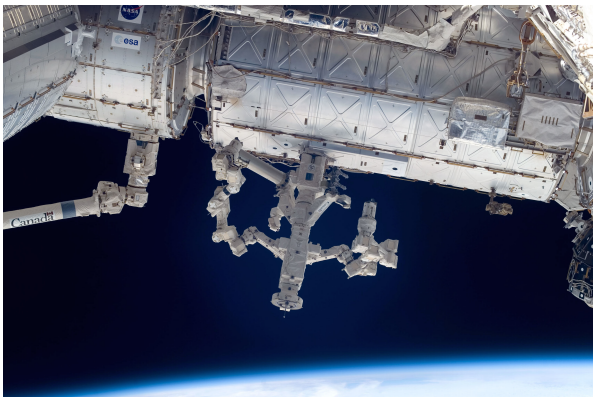
As mentioned above, the space robotics has one of the highest TRL among the servicing or debris removal technologies. The past missions described earlier demonstrated their benefits and many robotic arms are still in operation on the ISS. Moreover, some missions around the world are pushing further these fields of research to enforce for rendezvous and capture of non-cooperative objects in the coming years. A review of the actual robotic arms aboard the ISS is first made, and the main missions in preparation for debris removal or servicing are exposed. Again, some illustrations of the spacecraft concepts are given in Figure 1.9.

(a) Canadarm & Space Shuttle. Source: NASA

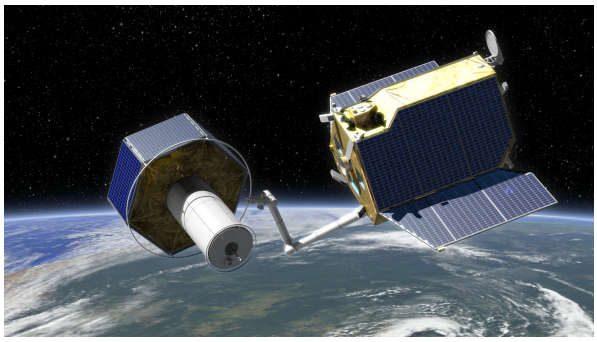


(b) ETS-VII mission. Source: JAXA

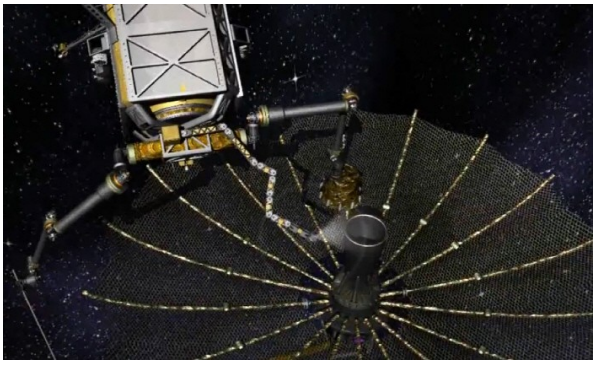
(c) Orbital Express mission. Source: DARPA



(d) DEXTRE robotic arm. Source: NASA



(e) TECSAS/DEOS mission. Source: DLR



(f) Phœnix mission. Source: DARPA

Figure 1.9: Most significant space robotics missions for servicing/debris removal

Table 1.1: Characteristics of the main space robotic arms

Name	Base	DoF	Length	Mass	Power	Year	Control
Canadarm[3]	Space Shuttle (68.6 t)	6	15.2 m	450 kg	1,000 W	1980	Manual
Canadarm2[4]	ISS (400 t)	7	17.6 m	1,800 kg	435 W	2006	Manual
Dextre[5]	ISS	15	2×3.5 m	1,662 kg	600 W	2008	Manual
JEMRMS[6]	ISS	12	12.2 m	970 kg	-	2008	Manual
ERA[7]	ISS	7	11.3 m	630 kg	475 W	2017?	Manual
ETS-VII[8]	Hikoboshi (2.5 t)	6	2.0 m	45 kg	80 W	1997	Ground
OE[9]	ASTRO (1.1 t)	6	3.3 m	71 kg	131 W	2006	Autonomous
DEOS[10]	Servicer (0.79 t)	7	3.2 m	40.5 kg	100 W	2018?	Autonomous

1.2.3.1 Robotic arms aboard the ISS

Among the most famous robotic arm brought up to the ISS, one can find the *Canadarm2*, its extension *Dextre*, the *Japanese Experiment Module Remote Manipulator System (JEMRMS)* as part of the Japanese module, the two Russian cranes *Strela* and the *European Robotic Arm (ERA)*, about to be launched next year. Their main characteristics in terms of DoF or mass and span have been compiled in Table 1.1.

It is worth noticing the difference in dimension between these arms and the ones embedded in the previous missions. They are designed to carry huge loads on long distance when assembling modules, so they result in heavy structures and long span. Moreover, since the ISS is an extremely massive base compared to them, they do not impact much its attitude when moving and their inertia is not a concern.

On the contrary, when the mass ratio between the arm and the base is more critical, the attitude control of the spacecraft is much more challenging. This is why the robotic arms of ETS-VII or OE were much lighter.

[3](Aikenhead et al., 1983)
[4]http://www.nasa.gov/mission_pages/station/structure/elements/mss.html
[5]https://www.nasa.gov/mission_pages/station/structure/elements/dextre.html
[6]http://iss.jaxa.jp/en/kibo/about/kibo/rms/
[7]http://www.esa.int/Our_Activities/Human_Spaceflight/International_Space_Station/European_Robotic_Arm
[8]https://directory.eoportal.org/web/eoportal/satellite-missions/e/ets-vii
[9]http://www.slideserve.com/patsy/orbital-express-a-new-chapter-in-space
[10](Rank et al., 2011)

1.2.3.2 Missions to come

Two main missions are supposed to be launched in the upcoming years: the German project for debris removal with *DEutsche Orbitale Servicing mission (DEOS)*, and the American Phœnix.

The first one is illustrated in Figure 1.9e and is also built on the idea of a common launch with both the chaser and target. This project was originally called *TEChnology SAtellites for demonstration and verification of Space systems (TECSAS)* and was a cooperation between the German, Canadian and Russian space agencies (Martin et al., 2005; Flores-Abad et al., 2014). Unfortunately, it suffered reorientations from its involved members in 2006, such that *Deutsches zentrum für Luft- und Raumfahrt (DLR)* decided to take the lead alone and came up with the DEOS project since 2009[11]. Its main goal is now to demonstrate an autonomous rendezvous, fly-around, capture and stabilization of a non-cooperative debris (Reintsema et al., 2010). The DLR agency already demonstrated a great expertise in space robotics with the *RObot Komponent Verification on ISS (ROKVISS)* project embedded on the Russian module of the ISS (Hirzinger et al., 2004). Moreover, most of the developed technologies are also jointly used for the servicing project *Orbital Life Extension Vehicle (OLEV)*, providing life extension for geostationary satellites. The DLR also validates intensively its technologies on the ground with the state-of-the-art robotic testbench *European Proximity Operations Simulator (EPOS)* wich allows for *Hardware-In-the-Loop (HIL)* experiments and rendezvous scenarios with satellite mockups (Boge et al., 2010; Boge and Ma, 2011).

The second mission of interest leans more toward on-orbit servicing, but it will push further many common technologies, like the robotic manipulation or the image processing. The Phœnix mission is built upon the legacy of two former missions. The original program was the *Spacecraft for the Unmanned Modification of Orbits (SUMO)*, initiated in 2002 to capture and service malfunctioning satellites in geostationary orbits, and not initially designed for servicing purposes. Then followed the *Front-end Robotics Enabling Near-term Demonstration (FREND)* project as part of the SUMO program, one of whose main contributions was the design of a multi-purpose arm, used on the last Phœnix mission (Obermark et al., 2007). The advances of the image processing paved the way for an autonomous capture of non-cooperative objects at low tumbling rate (Flores-Abad et al., 2014). On the actual design of the Phœnix mission, three arms are embedded on the servicer: two of them service the target, and the third one is dedicated to image acquisition and processing allowing for much more

[11] http://www.dlr.de/rmc/rm/en/desktopdefault.aspx/tabid-3825/5963_read-8759/

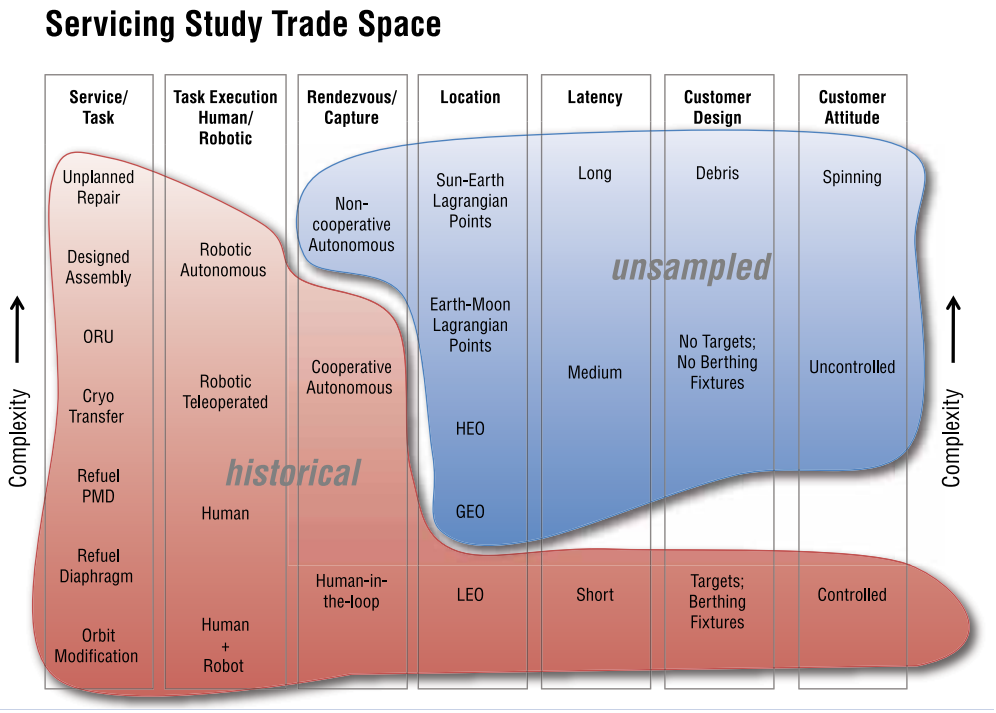


Figure 1.10: Balance between fulfilled and remaining challenges for on-orbit servicing. Source: NASA (NASA, 2010)

flexibility during the operations. The arms are direct evolutions of the FREND program. One improvement is the design of new tools used at the end-effector and above all the capacity to change autonomously the tool to perform multiple tasks with the same arm (Sullivan et al., 2015). Even if this last project is conducted in the scope of on-orbit servicing, its demonstration will create strong assets for space robotics, especially useful in the future debris removal missions.

1.3 Remaining Challenges

As a conclusion, this introductory part has presented the most promising technologies for active debris removal and on-orbit servicing. First, the orbiting space debris have been classified according to their size, their orbit, and their probability of collision. This ranking has allowed to exhibit a priority list for future removal missions, and to adapt the mission plan depending on the deorbitation method and the number of objects to treat.

Indeed, many concepts were presented to deorbit them, from the ground-based lasers to the capture devices. Nowadays, the two most promising technologies are the robotic arms and

the nets. Both emerge as the best compromises for the active debris removal, considering their maturity, their related risks, and the advantages they have. The focus has been mainly put on the space robotics, because of its versatility for servicing as well and its promising developments on the upcoming missions. A clear overview of the accomplished and remaining challenges is given in Figure 1.10 from the servicing standpoint.

The present introduction has also shown how autonomous rendezvous, fly-around and docking have been performed in the past. In addition, the robotic captures were mainly achieved by astronauts or by tele-operation, and always with cooperative targets. Therefore, an autonomous capture with a non-cooperative object is still an open problem and fuel the current researches. Such debris must be captured with tumbling rates of a few degrees per second (Lampariello, 2013; Bonnal et al., 2013), and most often without any visual marker nor grappling fixtures (Inaba and Oda, 2000; Friend, 2008). These last points are intensive axes of research, along with the flexible behavior of such manipulators. Indeed, their structure is optimized to minimize as much as possible the global mass at launch. These lightweight arms exhibit very low frequency modes, that can severely degrade the controller performances.

Going further in the technical details, the next chapter brings an overview of the current research topics on the modeling and control of such systems. Based on this critical analysis of the literature, the research objectives and the expected contributions of the thesis are given.

CHAPTER 2 LITERATURE REVIEW

This thesis aims at studying the modeling and the control of a space robot. The name "space robot" denotes a system composed of a "base" on which a "robotic manipulator" is embedded. The base includes all the usual components of a spacecraft: a rigid central body, with some appendages, like solar panels and antennas, and with an *Attitude and Orbital Control System (AOCS)*, including actuators and sensors. The robot itself is a system of connected and actuated bodies. It can be modeled by either rigid or flexible segments, and its joints have commonly one degree of freedom.

Therefore, the following review mixes both spacecraft and robotics theories. The emphasis is put first on the modeling and control of fixed-base robots with flexible members. Secondly, spacecraft dynamics is quickly covered to focus more on the modeling and control of space robots. In both sections, the usual control schemes and control strategies are also introduced. A section follows to address more specifically the guidance of a space robot aiming at capturing a tumbling debris. Then, the controller synthesis and analysis are developed for the purpose of flexible systems and for robust synthesis. Finally, the last part reviews the current means of validation and testing for space robotics.

2.1 Fixed-Base Robotics

Fixed-base robots are already thoroughly covered through reference books like (Spong et al., 2006; Craig, 1989; Featherstone, 2008). Nevertheless, there is still room for improvement in research avenues, such as flexible modeling, obstacle avoidance or localization and mapping (Garcia et al., 2007). For example, service robotics is perceived as a high potential application in the near future, while medical robots are already employed [1]. It is worth recalling that many of them have their technical roots in space robotics, in terms of high dexterity effectors and vibration reductions [2]. The present section gives the key references for fixed-base robotics, with a focus on flexible modeling and control. This last point is crucial for space manipulators due to their extreme slenderness and lightness.

For both sections on rigid and flexible robotics, the modeling is covered through the kinematics, kinetics, dynamics and simulation aspects.

[1] http://mdacorporation.com/isg/robotics-automation/commercial-systems/medical
[2] http://www.asc-csa.gc.ca/eng/canadarm/robotic.asp

2.1.1 Rigid modeling & control

Rigid robots consists in the interconnection of rigid bodies. Their description is readily obtained by applying the corresponding constraints of motion on their free kinetics and dynamics.

Kinematics Rigid kinematics was firstly based on the *Denavit Hartenberg (DH) parameters* (Denavit and Hartenberg, 1955). Four scalar quantities per segment are sufficient to fully describe the position and attitude of any point along the robotic arm. Using the DH formalism, transformation matrices are defined to express the local coordinates of a point in the inertial frame, linked to the base. This is mainly used to express the end-effector position. Unfortunately, these parameters turned to be ambiguous when dealing with tree-type systems (Khalil, 2010). *Modified Denavit Hartenberg (DH) parameters* were thus introduced to overcome this limitation by W. Khalil and J. F. Kleinfinger in (Khalil and Kleinfinger, 1986). A similar set of four parameters is proposed to build the transformations by a different sequence of rotations and translations.

Among these parameters, three are constant for a one degree-of-freedom segment (i.e., allowed to move along only one direction). The variable ones form the *arm configuration*, denoted $\mathbf{q} \in \mathbb{R}^n$ with n being the number of rigid DoF. In this case, it is also equal to the number of segments. With this notation, the kinematic relation giving the end-effector position and attitude w.r.t. the inertial frame reads:

$$\mathbf{x}_{\mathrm{E}} = \mathbf{f}(\mathbf{q}) \tag{2.1}$$

Kinetics When deriving this kinematic relation w.r.t. time, the angular and linear speeds of this point are obtained and merged into the twist $\mathbf{t}$. The **Jacobian** matrix of the manipulator $\mathbf{J}_{\mathrm{m}}$ comes out to express the end-effector twist as:

$$\mathbf{t}_{\mathrm{E}} = \mathbf{J}_{\mathrm{m}}(\mathbf{q})\,\dot{\mathbf{q}} \tag{2.2}$$

This Jacobian matrix depends only on the arm configuration $\mathbf{q}$. When it looses its full rank, the corresponding configuration is called a *kinematic singularity* (Craig, 1989; Spong et al., 2006). For a fixed-based manipulator, these configurations occur when two axes of rotation are aligned, or when the workspace boundary is reached. It means that the effector can no longer move along the directions corresponding to the zero singular values of the Jacobian.

Dynamics The previous kinetic relation can be applied for the segments *Center of Mass (CoM)*. Their position and speed are expressed in terms of the arm configuration, in order to derive the global kinetic energy or the segment accelerations. The dynamics of the manipulator can be obtained by two main approaches (Spong et al., 2006).

- **Local:** The free dynamic equations of each segment are used in the local approach. The constraints of motion are then applied to take into account that each body is interacting with its direct neighbors. This approach is called *Newton-Euler*, because the translational dynamics is given by the classic Newton laws, and the rotational ones by the Euler equation. Using the arm configuration and its time derivatives, the position, speed and acceleration of the segment CoM are computed to derive the resulting efforts applied along its axis of motion. For each segment, the dynamic equation reads:

$$\forall i = 1 \dots n, \qquad \mathbf{M}_i \begin{bmatrix} \dot{\boldsymbol{v}}_i \\ \dot{\boldsymbol{\omega}}_i \end{bmatrix} + \mathbf{h}_i(\boldsymbol{v}_i, \boldsymbol{\omega}_i) = \begin{bmatrix} \mathbf{f}_i \\ \mathbf{n}_i \end{bmatrix} \tag{2.3}$$

  with $(\boldsymbol{v}_i, \boldsymbol{\omega}_i)$ the segment linear and angular velocities at a given reference point, $\mathbf{M}_i$ the segment mass matrix, $\mathbf{h}_i$ the vector of nonlinear efforts (or Coriolis and centrifugal vector), and $(\mathbf{f}_i, \mathbf{n}_i)$ the forces and torques applied on it. The corresponding generalized effort τ_i is obtained by extracting from $(\mathbf{f}_i, \mathbf{n}_i)$ the resulting effort applied along the axis of motion.

- **Global:** The manipulator is now considered as a whole. By iterating on each segment, the kinetic and potential energies are computed to derive the global Lagrangian. The corresponding *Lagrangian equations* are then derived by considering the previous DH parameters $\mathbf{q}_\mathrm{r}$ as *generalized coordinates*. Similar for most mechanical systems (Spong et al., 2006; Craig, 1989), the closed-form equation of motion is usually written as:

$$\mathbf{D}_\mathrm{r}(\mathbf{q}_\mathrm{r})\, \ddot{\mathbf{q}}_\mathrm{r} + \mathbf{h}_\mathrm{r}(\mathbf{q}_\mathrm{r}, \dot{\mathbf{q}}_\mathrm{r}) = \boldsymbol{\tau}_\mathrm{r} \tag{2.4}$$

  with $\mathbf{D}_\mathrm{r}(\mathbf{q}_\mathrm{r})$ the global mass matrix, $\mathbf{h}_\mathrm{r}(\mathbf{q}_\mathrm{r}, \dot{\mathbf{q}}_\mathrm{r})$ the global vector of nonlinear effects, and $\boldsymbol{\tau}_\mathrm{r}$ the generalized efforts (i.e., force/torque depending on the joint type).

The previous dynamic schemes are used in two ways: *inverse* and *forward* dynamics. Inverse dynamics computes the joint efforts $\boldsymbol{\tau}_\mathrm{r}$ resulting from a given arm configuration $(\mathbf{q}_\mathrm{r}, \dot{\mathbf{q}}_\mathrm{r}, \ddot{\mathbf{q}}_\mathrm{r})$. For example, it may be used for open-loop control, by computing the efforts to apply at the joints, $\boldsymbol{\tau}_{des}$, in order to produce a desired trajectory, given by $(\mathbf{q}_{des}, \dot{\mathbf{q}}_{des}, \ddot{\mathbf{q}}_{des})$. On the other hand, forward dynamics is used to compute the resulting accelerations $\ddot{\mathbf{q}}_\mathrm{r}$ when some given efforts $\boldsymbol{\tau}_\mathrm{r}$ are applied. To do this, they need to invert the global mass matrix and appeal to sophisticated simulation schemes.

Simulation Efficient recursive algorithms are proposed for the Newton-Euler approach in (Walker and Orin, 1982), and for the Lagrangian one in (Hollerbach, 1980). They reach similar $\mathcal{O}(n)$ computational complexity, varying thus linearly with the number of segments. But the recursive Newton-Euler scheme proved to be more popular than the Lagrangian one, because it was slightly more efficient in terms of multiplications according to (Featherstone and Orin, 2000). The main Newton-Euler methods are summarized in (Featherstone, 2008). Among them, one finds the *Recursive Newton-Euler Algorithm (RNEA)* for inverse dynamics, the *Composite Rigid Body Algorithm (CRBA)* for computing the global mass matrix, and the *Articulated-Body Algorithm (ABA)* for forward dynamics.

Already mentioned above, the first one is built upon a recursive computation of the generalized efforts $\boldsymbol{\tau}$, by using explicitly the internal efforts. Presented in (Walker and Orin, 1982), the starting point of CRBA is to notice that an inverse dynamics algorithm $\texttt{InvDyn}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ can be used to efficiently obtain the mass matrix. Indeed, the nonlinear effects are quadratic in the generalized speeds (Spong et al., 2006), such that the i^{th} column of the mass matrix is given by $\texttt{InvDyn}(\mathbf{q}, \mathbf{0}, \mathbf{e}_i)$, denoting by $\mathbf{e}_i$ the i^{th} basis vector. More physical meaning is given in (Featherstone and Orin, 2000), showing that the algorithm actually evaluates recursively the inertia matrix of a composite system rooted at the current joint. This system is made of the rigidly connected subsegments emerging from this joint. Finally, the ABA algorithm developed by R. Featherstone is one of the most efficient methods for forward dynamics. Without the need to compute the whole mass matrix, it only inverts a reduced inertia matrix for each joint, resulting in a linear $\mathcal{O}(n)$ complexity.

A similar approach is brought by the *Decoupled Natural Orthogonal Complement (DeNOC)* algorithm. Originally, J. Angeles and S. K. Lee introduced the *Natural Orthogonal Complement (NOC)* method to eliminate the internal efforts from the equations of free dynamics (Angeles and Lee, 1988). Based on the Kane's formulation (Parsa, 2007), it reduces the $6n$ equations of motion for free rigid bodies to only n. Indeed, since the segments are constrained to translate/rotate only along their joint axis, the linear dependency between twists and joint speeds in (2.2) is extended to every segment's CoM. Gathering these twists and speeds into $\mathbf{t}$ and $\dot{\mathbf{q}}$, this relation is extended as follows:

$$\mathbf{t} = \mathbf{N}(\mathbf{q})\,\dot{\mathbf{q}} \tag{2.5}$$

with $\mathbf{N} \in \mathbb{R}^{6n \times n}$. The equations of motion are then reduced by pre-multiplying them by $\mathbf{N}$, in the form of Kane's equations. S. K. Saha pushed further the NOC approach by expressing $\mathbf{N}$ as the product of two matrices: $\mathbf{N} = \mathbf{N}_l\,\mathbf{N}_d$, where $\mathbf{N}_l \in \mathbb{R}^{6n \times 6n}$ is a lower triangular matrix, and $\mathbf{N}_d \in \mathbb{R}^{6n \times n}$ is a block diagonal matrix. This decomposition leads to

the DeNOC algorithm and provides analytical and compact expressions for the manipulator dynamics (Saha, 1999). The mass matrix is obtained in the same way as the CRBA (Saha, 1997). A *Recursive Gaussian Elimination (RGE)* is even proposed to invert the manipulator mass matrix without the need to compute it explicitly. In that sense, forward dynamics is solved in a same way as the ABA algorithm does. All steps are detailed for rigid bodies through (Mohan and Saha, 2007), and a state-of-the-art about the DeNOC applications is available in (Saha et al., 2013). One of its major advantages lies in its immediate extension to flexible segments, as explained in the following.

Alternative approaches are also presented in the literature. G. Rodriguez reveals an interesting analogy between the rigid forward dynamics and the Kalman filtering, and Bryson-Frazier smoothing equations all together (Rodriguez, 1987). He pushes further the idea by building the *Spatial Operator Algebra* for modeling, control and even path planning of rigid manipulators (Rodriguez et al., 1991). An operational-space model is also proposed by O. Khatib in (Khatib, 1987), in order to explicit the dynamics w.r.t. the cartesian coordinates of the effector. This approach is very straightforward to derive a cartesian path planning.

Control The control of robotic arms may be classified into the usual linear and nonlinear methods. Starting with the most simple scheme, one can use mere *Proportional-Derivative (PD)* controllers at each joint (Kawamura et al., 1988). It is shown how the Derivative action can always be set high enough to track any reference trajectory, but at the expense of high command levels. The main advantage of these methods is their low computational cost, while the main drawback is the inevitable drop in performance, due to the limited capabilities of the actuators. For example, H_∞ methodology is a widespread tool to tune linear controllers. It is given in more details into Section 2.4, dedicated to control.

Regarding the nonlinear approach, classic controllers are given by the transposed and inverse Jacobian schemes (Craig, 1989). They are illustrated, respectively, in Figure 2.1a and Figure 2.1b. In both schemes, the reference signal is directly the desired effector trajectory. In order to avoid kinematic computations inside the feedback loop, it is far better to provide inertial measurements of the effector position. For example, a ground-based camera is a common sensor for this kind of scheme (Umetani and Yoshida, 1989a). As mentioned above, the Jacobian matrix is configuration-dependent and has to be updated as the robotic arm is moving. To do this, efficient algorithms are given in (Spong et al., 2006).

Another classic scheme is the computed-torque control, which is nothing more than a feedback linearization from the point of view of nonlinear control theory (Khalil, 2001). It decouples the manipulator dynamics by using the inverse dynamics problem in (2.4), and then uses a classic PD or *Proportional-Integral-Derivative (PID)* controller on the independent states.

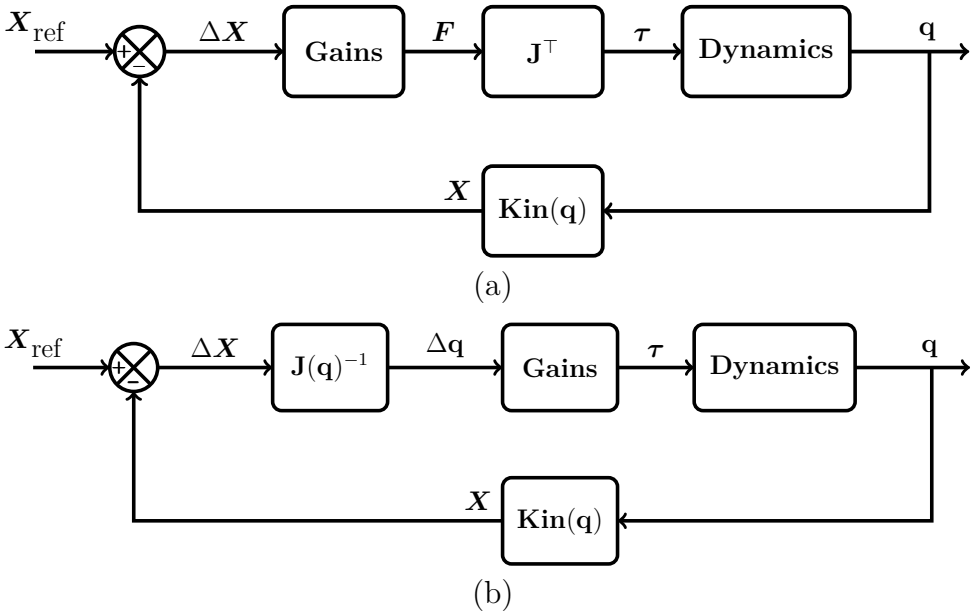
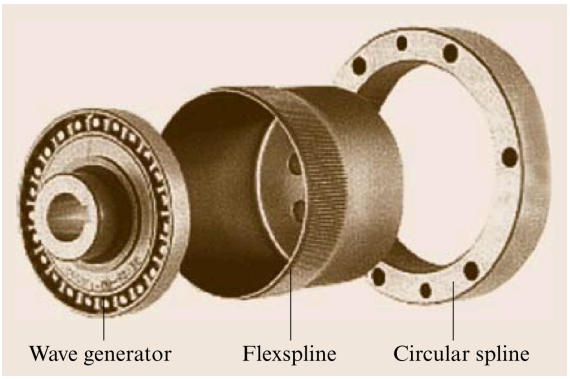


Figure 2.1: Jacobian-based control for fixed-base robots;
(a) Transposed Jacobian structure; (b) Inverse Jacobian structure.

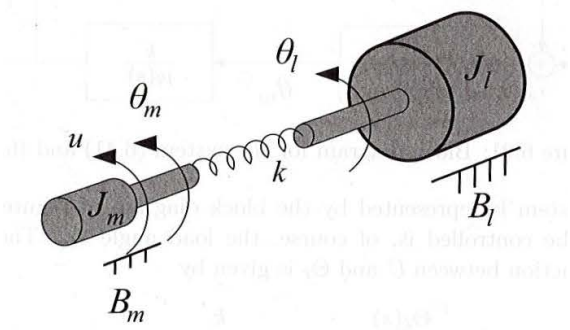
A real-time implementation is proposed in (Khosla and Kanade, 1989), and compares with the performances of the previous PD controllers for two computed-torque implementations: one with the Coriolis and centrifugal terms, and another one without them. It turns out that these terms are necessary to reach a good trajectory tracking, but that the required command often reaches the saturation level.

Finally, a last method of control which is worth mentioning is the *Operational Space Formulation* developed by O. Khatib. Based on the effector dynamics in the cartesian space, hybrid force/motion control is proposed to ensure a constrained tool motion when a contact has to be maintained with the environment (Khatib, 1987). D. N. Nenchev goes further in (Nenchev, 2013), by using the *Reaction Null Space* control method to perform a force/motion control less sensitive to the kinematic singularities. This method is described in more detail through the space robot section.

Reminding that the final goal of the thesis is to model and to control lightweight space robots, the flexible counterparts of the previous methods are presented in the next section. Emphasis is given to the flexible extension of rigid modeling techniques and to the new issues arising to control these flexible manipulators.



(a) Source: (De Luca and Book, 2008)



(b) Source: (Spong et al., 2006)

Figure 2.2: Elastic joints of a robotic arm;
(a) Harmonic drive mechanisms; (b) Schematic behavior of in-joint elasticity.

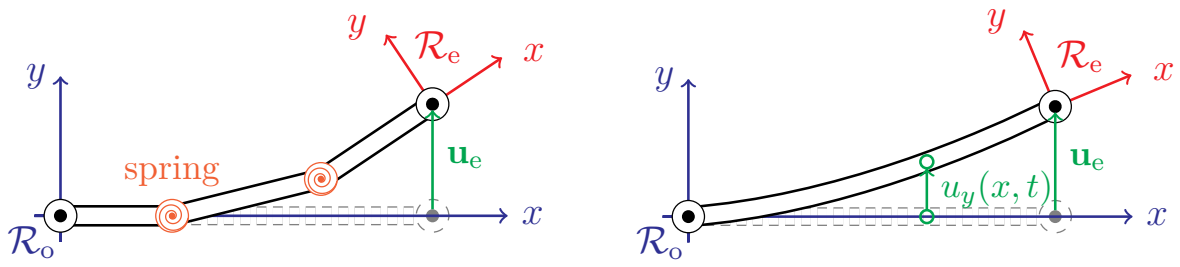
2.1.2 Flexible modeling and control

Flexible systems are not common in industrial robotics, which is usually based on heavy and stiff segments. Joint flexibility is the main source of concern with the use of gearboxes. The motion of large payloads or high-speed trajectories can also induce flexible deflections, that lower performances and may lead to high vibrations if uncontrolled.

For on-orbit systems, and especially embedded flexible manipulators, this issue is even worsened by the segment flexibility. In addition, experiments are not always possible because of their prohibitive cost or because of the lack of gravity compensation facilities. Their simulation is thus necessary, but becomes more and more cumbersome as the number of degrees-of-freedom increase. Indeed, their dynamics lies on high order differential equations. An introductory work is given in (Piedboeuf, 1992), while A. A. Shabana in (Shabana, 1997) and S. K. Dwivedy and P. Eberhard in (Dwivedy and Eberhard, 2006) propose two thorough reviews on the field. Eventually, A. De Luca also presents a control-oriented state of the art in (De Luca and Book, 2008).

To distinguish both sources of deformation, the torsional joint deflection is usually called *elasticity*, while *flexibility* rather stands for the segment one (Piedboeuf, 1992).

In-joint elasticity As emphasized in (Flores-Abad et al., 2014), revolute joints often contain a harmonic drive gearbox. Thanks to a coaxial assembly, it increases the torque capacity while remaining very compact. The motor shafts also have to endure high inertia when subsegments are moving at high speed or when a load is grasped by the end-effector. It creates a torsional effect which generates joint oscillations around the desired angle. A common



(a) Lumped flexibilities (b) Distributed flexibility

Figure 2.3: Flexible links of a robotic arm;
(a) Lumped flexibilities with *virtual joints*; (b) Assumed modes method with distributed flexibility.

modeling approach is to introduce a passive joint with a torsional spring of relatively low stiffness (Nicosia et al., 1981; Spong, 1992). An example of the joint mechanisms and the corresponding flexible model are given in Figures 2.2a and 2.2b. Control issues for this type of in-joint flexibility has been widely addressed in the literature (Spong et al., 2006; De Luca and Book, 2008). The use of PD controllers is investigated and the control scheme is improved by feeding back the motor angle and velocity instead of the link ones.

Segment flexibility For the modeling of long flexible segments, many different approaches exist and are well summarized through (Dwivedy and Eberhard, 2006). One simple but effective idea is to cut them into many rigid sub-segments driven by virtual passive joints, as shown in Figure 2.3a. Many names are given to this model, like the *fictitious joint approach* (Alazard and Chrétien, 1992) or the *finite segment approach* (Shabana, 1997). Since the same techniques presented above can be used for the rigid-body modeling, it reduces drastically the computational burden in simulation. Still, proper values of stiffness and damping for the passive joints have to be used, using either an energy equivalence (Chrétien et al., 1985) or an identification with an experimental setup (Yoshikawa and Hosoda, 1991). These models provide a suitable modal analysis for a preliminary control design, but they require too many virtual joints to remain valid in simulation.

A more accurate modeling is obtained with the *Assumed Mode Method (AMM)* illustrated in Figure 2.3b. The flexible deformation is described by a set of modes, given by the product of a time amplitude by a spatial function defined over the whole segment. The computation of these modal shapes is done using different structural beam models (Timoshenko, 1955). The corresponding *Partial Differential Equation (PDE)* to compute the eigenfunctions describing each mode are gathered in (Meirovitch, 2001). The most common one is the Euler-Bernoulli

model governing the dynamics of a slender beam. Transverse shear and rotary inertias are neglected, and the beam section is supposed to remain planar and orthogonal to the main axis (Piedboeuf, 1992). Since the sub-segments have non-negligible masses and inertias, the clamped-loaded boundary conditions must be used instead of clamped-free ones (Oakley and Cannon, 1989; De Luca and Siciliano, 1991). A more precise and intricate model is also available through the Timoshenko beam theory, if one needs to account for shear deformation and structural visco-elasticity (Loudini et al., 2006).

A comparison is given in (Alazard and Chrétien, 1992) between the fictitious joint approach, the assumed modes method, and a last one based on assumed polynomial shapes. The study is carried out on the modeling of a loaded flexible manipulator from a control point of view. It is shown that the three methods yield similar results concerning the system modal frequencies but it does not investigate their behavior in simulation. A flexible model based on polynomials and Rayleigh-Ritz expansion is also proposed in (Nicosia et al., 1996). The link deformation is approximated by these polynomial expressions and validated on an experimental setup. Similar works aim at building models for control synthesis. The second order relations between flexible variables and strain inside the beam are used to build a two port model in (Murali et al., 2015). Still using polynomials to describe the flexible deflections, the accelerations and the interaction forces are propagated through each element to build the linear model of any multi-body structure.

A modeling method is also available for more complex shapes: the *Finite Element Method (FEM)*. Based on a refined discretization of the beam for an improved accuracy, a great number of state variables are necessary in simulation. Since the resulting computation time increases, it rather serves as a reference case to evaluate the validity of a given flexible model (Dwivedy and Eberhard, 2006). The AMM appears to be a good candidate to serve both in the control design and in the simulation of flexible elements.

Kinematics/Kinetics Whatever the flexible model, the manipulator kinematics can still be expressed using the usual 4×4 transformation matrices of rigid robotics (Book, 1984). They are augmented with the flexible deflections, expressed as the summation of modal shapes. Nevertheless, the influence of the beam bending and torsion must be carefully considered to update the global rotation matrix and the flexible angular rate (Damaren and Sharf, 1995). An improved AMM kinematics framework is also proposed in (Oakley and Cannon, 1989) to account for the link foreshortening when it is bending. Indeed, the classic AMM assumes that the beam has always the same length.

Dynamics The derivation of flexible dynamics is mainly based on the Lagrangian equation. A recursive approach is proposed in (Book, 1984) to compute the speed and acceleration of any point along a flexible member, and then to build the mass matrix based on the Lagrangian equations. This recursion also allows for the automatic derivation of dynamic equations, using symbolic languages (De Luca et al., 1988). A complete example for a planar flexible manipulator with two segments is given with the AMM in (De Luca and Siciliano, 1991), and with the FEM in (Usoro et al., 1986).

A general mathematical framework is introduced in (Boyer and Glandais, 1999; Boyer et al., 2010) to derive the flexible kinematics and dynamics by the *equilibrium principle* rather than a Lagrangian approach. They come back to the definition of the Eulerian and Lagrangian formalism to express, respectively, the rigid and flexible motions of a segment. This approach is similar in every way to the *quasi-coordinates* concept of L. Meirovitch in (Meirovitch, 1991). The main idea is still to use the linear and angular speeds to describe the rigid-body motion of the segment, while the flexible motion is parametrized by the generalized flexible coordinates (i.e., the mode amplitudes for the AMM approach).

An alternative modeling technique is available through (Vakil et al., 2012). The flexible behavior of joints and segments is decoupled into two sub-models: one with flexible segments and rigid joints, and a second with only flexible joints. The work is based on the Lagrangian equation of a single flexible beam mounted on a moving base. It avoids thus the intricate computation of the global Lagrangian to derive the closed-form dynamic equation in (2.6). Unfortunately this method is developed for planar robots only.

In the general case, flexible dynamics is written in the same way using either the AMM or the FEM:

$$\mathbf{D}_{\mathrm{f}}(\mathbf{q}_{\mathrm{f}})\,\ddot{\mathbf{q}}_{\mathrm{f}} + \mathbf{h}_{\mathrm{f}}(\mathbf{q}_{\mathrm{f}}, \dot{\mathbf{q}}_{\mathrm{f}}) + \mathbf{B}_{\mathrm{f}}\,\dot{\mathbf{q}}_{\mathrm{f}} + \mathbf{K}_{\mathrm{f}}\,\mathbf{q}_{\mathrm{f}} = \mathbf{Q}\,\boldsymbol{\tau}_{\mathrm{r}} \tag{2.6}$$

where the subscript "f" denotes the flexible version of previous matrices. The generalized coordinates $\mathbf{q}_{\mathrm{f}} = \begin{bmatrix} \mathbf{q}_{\mathrm{r}}^{\top} & \boldsymbol{\delta}^{\top} \end{bmatrix}^{\top}$ are augmented with the flexible coordinates $\boldsymbol{\delta}$. The mass matrix $\mathbf{D}_{\mathrm{f}}$ and the vector of Coriolis and centrifugal terms $\mathbf{h}_{\mathrm{f}}$ are both augmented to account for the flexible dynamics. Finally, $\mathbf{K}_f$ and $\mathbf{B}_f$ are the flexible stiffness and damping matrix, and $\mathbf{Q}$ is the input matrix.

Simulation An adapted Newton-Euler scheme proves to be very efficient to simulate the flexible dynamics. Using the dynamic model of a single segment, F. Boyer and P. Coiffet extend the rigid scheme to flexible elements (Boyer and Coiffet, 1996). They added the flexible coordinates dynamics to the classic Newton/Euler laws of rigid-body motion. The full algorithm is available through (Boyer et al., 2010; Benosman et al., 2002). A generalization

is also proposed for parallel robots in (Briot and Khalil, 2013).

The rigid NOC approach is also generalized for flexible members by Cyril (Cyril, 1988). In the same way, the DeNOC approach is extended to flexible links by A. Mohan and S. K. Saha in (Mohan and Saha, 2009). Using the Lagrangian approach, analytical expressions are derived for the global mass matrix $\mathbf{D}_\mathrm{f}$ and the nonlinear vector $\mathbf{h}_\mathrm{f}$. The main advantage is that the *Recursive Gaussian Elimination (RGE)* technique extends for flexible forward dynamics in a straightforward way. It is shown in (Mohan and Saha, 2009) that this algorithm leads to one of the most efficient solutions to compute the forward dynamics.

One major drawback of these methods is the general formulation of dynamics in terms of integrals over the whole segment. Indeed, the Lagrangian equations are derived from the kinetic energy integrated over the whole manipulator. Since time-varying terms appear in these integrals, they should be updated on-line during the simulation and would increase greatly the computational burden. A step forward is done by C. Damaren and I. Sharf in (Damaren and Sharf, 1995) to reduce the dynamic model in a coherent way. Approximate models are built according to the number of inertial and geometric terms neglected in the integrals. They are eventually compared in simulation to choose the most relevant terms.

Control The control of flexible manipulators is now reviewed as a conclusion for the section on fixed-base robotics. As mentioned earlier, classic PD controllers provides good performance for a collocated feedback, with the motor angle and velocity (Spong et al., 2006). A good insight into the impedance control of elastic-joint robot is given in (Ott et al., 2008). Using a joint torque feedback, they develop two controllers for gravity compensation and impedance control. Both the joint-space control and the cartesian-space one are considered using the Jacobian matrix. Proofs of system passivity and stability with the gravity compensation are provided for both controllers. They are also validated with the manipulators of the DLR.

Regarding flexible-link robots, input or command shaping can be used, but the active control techniques seem more suited to counteract the loss of precision due to the flexible deflections (De Luca and Book, 2008). One major challenge for these systems is the presence of zeros in the right half plane, also said to be *non-minimum phase*. Indeed, the control of the effector motion with distributed flexibility leads to a non-collocated control scheme (Kwon and Book, 1994). It is thus impossible to implement a direct inversion in practice, due to the corresponding unstable terms. To overcome this limitation, approaches based on the transfer function analysis are introduced in (Kwon and Book, 1994; Benosman et al., 2002).

Using the linear flexible dynamics, inverse dynamics are solved in the time domain, where the unstable exponential terms appear clearly. By canceling their value, a stable scheme is proposed for end-effector trajectory planning. If the loop is closed to perform an input-output linearization, the controller stability must be carefully investigated (De Luca et al., 1988). This technique is used to design a nonlinear controller, and it is applied on a planar two-link robot with in-joint elasticity and link flexibility. They showed the presence of unobservable states in closed loop. The dynamics of these latter, called *sink dynamics*, must be studied to settle whether it would destabilize the system or not.

Considering the system dynamics, flexible models are known to be very large in size. In order to reduce the controller complexity, a singular perturbation approach is presented in (Siciliano and Book, 1988). Global flexible dynamics are split between slow rigid modes and fast flexible ones. A composite controller is proposed with a slow term controlling the rigid motion, and a fast one compensating for the flexible deflections. Its validity is proven in simulation on a single flexible link example.

Linear controllers are also investigated in (Oakley and Cannon, 1989). An experimental setup at the Stanford University is used to validate the controller tuning. A nonlinear estimator is used to predict the effector location, and to compare it with a reference trajectory. A *Linear Quadratic Regulator (LQR)* is then synthesized on the resulting error. For trajectory tracking, this end-point control turns to be much more efficient than a traditional collocated control with PD joint controller. An alternative approach is proposed in (Krauss, 2006) with the *Transfer Matrix Method*. Based on the linear state-space equations, this frequency domain model is built upon the propagation of static positions and efforts from one end of the beam to the other. Control-oriented models are derived to perform the controller synthesis, and a pole-placement optimization is introduced.

As mentioned above, passivity is also a powerful framework to get strong stability properties for the closed-loop system. To preserve it, the effector deflection can be weighted to ensure the passivity with the control torque (Saad, 2003). The system output is taken as an intermediate deflection along the manipulator. In order to keep describing faithfully the deflection, the location of this output is as close as possible from the tip. The closed-loop passivity is then investigated with PD controllers and the optimal sensor location is obtained to guarantee it. To go further, a time-domain passivity control including a trajectory planner is proposed in (Ryu et al., 2003). The energy flow going through the feedback controller and the flexible manipulator is described in terms of joint velocities and torques. A *passivity observer and controller* are designed and validated in simulation.

Furthermore, the controllability of flexible modes is not always ensured depending on the arm configuration. The controllability matrix related to the flexible states must be examined to find the singular configurations in joint and cartesian space (Lopez-Linares et al., 1997). When the manipulator reaches these configurations, the flexible modes cannot be damped nor controlled by the joint torques because the controllability matrix loses its full rank. This phenomenon is a strong limitation for the controller performances in 3D motion.

The dynamics and the simulation of flexible robots have been investigated throughout the last sections. The modeling of a space robot is now considered with the addition of a moving and actuated base. The additional challenges that arise are covered in each of the previous fields, from kinematics and dynamics to control strategies.

2.2 Space Robotics

When considered as a point mass, the coarse behavior of a space system is governed by the orbital dynamics and by the disturbances coming from the harsh space environment. But due to the multi-body structure of a space robot, the point mass hypothesis no longer stands and the modeling must be refined to encompass all the interactions between the appendages and the rigid base. Their complex attitude dynamics lie on a nonholonomic relation that translates the disruptive effect of an appendage motion onto the base spacecraft.

In the sequel, orbital dynamics are quickly reviewed in Section 2.2.1 and more efforts are put on the dynamics of space robot in Section 2.2.2, where they are considered as tree-type systems in weightlessness conditions. A great number of control schemes are available depending on the constraints considered. One may ensure a pointing performance of the spacecraft, or the following of a given trajectory by the robotic manipulator. The whole Section 2.2.3 is thus dedicated to review these schemes.

2.2.1 Orbital mechanics

The general behavior of the satellites on orbit is described through reference books like (Hughes, 1986) and (Chobotov, 1991). When reduced to point masses, their trajectory follows an orbit around the Earth, and is described by a set of six parameters illustrated in Figure 2.4. They are called the *Keplerian elements*, and read as follows for an elliptical orbit:

- **Plane of the orbit**: the Right Ascension of the Ascending Node (RAAN) Ω and the inclination i ;

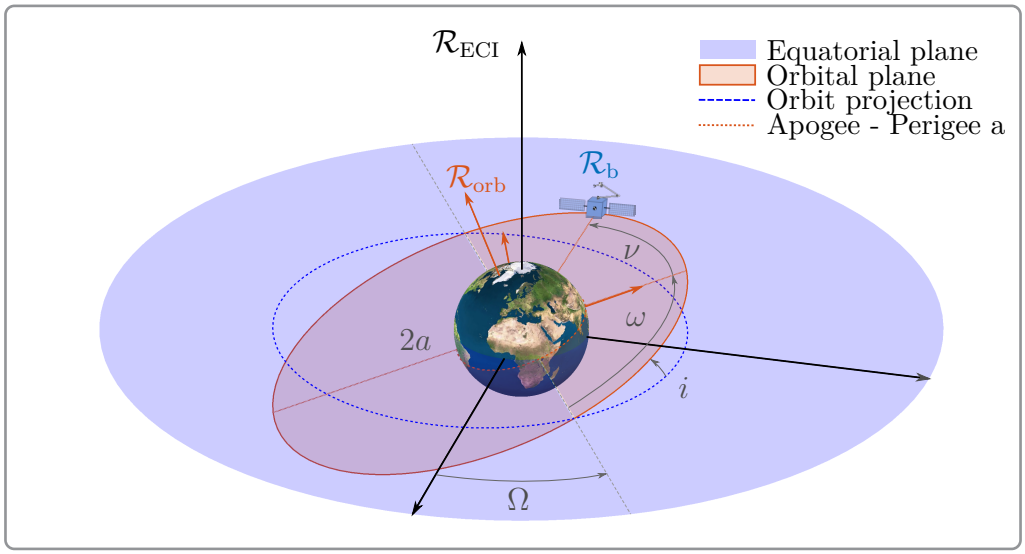


Figure 2.4: Keplerian elements of an orbit

- **Shape of the orbit**: the eccentricity e and the semi-major axis a ;
- **Orientation of the orbit**: the argument of the perigee ω ;
- **Satellite position**: either the mean anomaly M, the eccentric anomaly E, or the true anomaly ν.

Around this linear motion, the spacecraft attitude can be described by different sets of parameters. For rigid bodies, various representations may be used to give the angular orientation w.r.t. the Earth frame: Euler angles, Euler parameters, axis/angle parameterization, quaternions or even direct cosine matrices (Hughes, 1986). Based on four variables, the Euler parameters or the quaternions are often preferred for their stability in simulation and the avoidance of singular orientations. However, the Euler angles are easier to handle from a physical viewpoint, and they are a common output of the actual sensors.

To reach and maintain the desired orbit, a set of actuators, sensors and on-board processors are embedded on the spacecraft. They are referred to as the *Attitude and Orbital Control System (AOCS)* (Wertz et al., 2011). Actuators are used to apply forces and torques on the spacecraft, depending on the commands sent by the control computers. Among the most common ones, there are thrusters, reaction and momentum wheels, *Control Moment Gyroscope (CMG)* and magnetic torquers. On the other hand, the sensors determine the body attitude and angular rate w.r.t. inertial elements, in order to provide these data to the controller. They are mixing star trackers, sun sensors, GPS receivers, gyroscopes and magnetometers.

Some of these elements are spinning, like the reaction wheels or CMGs, so they influence greatly the spacecraft dynamics. The linear modeling of a satellite with these two spinning actuators is investigated for control purposes in (Alazard et al., 2008). Flexible solar panels are also a great source of disturbance. They must be included in the control design and synthesis when their lowest frequencies wrap around the controller bandwidth (Guy et al., 2014). More details are provided on the flexible issues arising for the spacecraft control in Section 2.4.2.

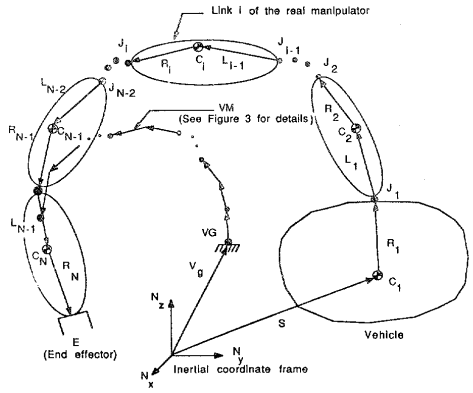
The nonlinear dynamics and control of the orbiting multi-body systems is covered in the next section. The focus is put more specifically on space robots, which contain actively controlled and chain-like appendages.

2.2.2 Space robot dynamics

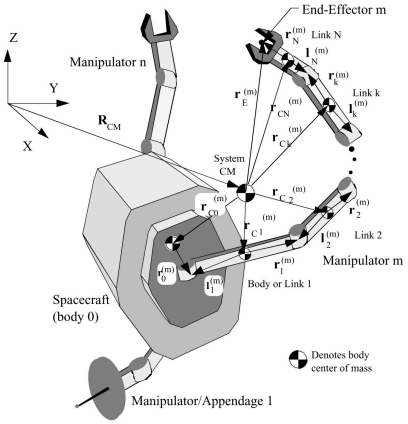
As mentioned in the introduction, servicing tasks and debris removal missions boil down to similar tasks with a robotic arm. A thorough review over the kinematics, dynamics, control and path planning of space robots is given in (Flores-Abad et al., 2014). The cases of spacecraft with either a single or multiple manipulators are equally treated. Dedicated reviews are also available for the kinematics and dynamics in (Dubowsky and Papadopoulos, 1993), for the control strategies in (Yoshida and Wilcox, 2008; Moosavian and Papadopoulos, 2007a) and for the path planning in the introduction of (Aghili, 2012). From a broader point of view, space applications cover a wide variety of systems, from the humanoïd and mobile robotics to the classic chain-like manipulators embedded on a spacecraft (Yoshida, 2009).

In the next sections, spacecraft with single robotic manipulators are emphasized to avoid the case of closed mechanical chains, which may occur for a multiple-arm system grasping a target (Hu and Vukovich, 1997). This underlying hypothesis drives the review on modeling, control and path planning of space robots to capture tumbling objects. By making this choice, the modeling of tree-type systems apply easily, and the case of human-like robots provides efficient algorithms (Shah et al., 2012a). Indeed, these latter also exhibit a moving base, that is usually not actuated, unlike the space robot.

Kinematics In 1987, R. W. Longman *et al.* proposed to describe the space robot with a kinematic scheme derived from the classic Newton-Euler algorithm for fixed-base robots (Longman et al., 1987). In order to compute the inertial location of the effector, the base motion is introduced as an initial condition in the kinematic loop. When performing the second recursion from the effector to the base, the disturbances resulting from the manipulator



(a) Source: (Vafa and Dubowsky, 1987)



(b) Source: (Papadopoulos and Moosavian, 1994)

Figure 2.5: Kinematics of space robots.
(a) *Virtual Manipulator* concept; (b) *Barycentric Vector* approach.

motion is computed and compensated by a feedforward control. The same kind of Newton-Euler algorithm is extended to multiple arms in (Carignan and Akin, 2000). If the generalized coordinates of the appendages are gathered in $\mathbf{q}$ and if $\mathbf{x}_\mathrm{b}$ denotes the base coordinates, the effector position is now influenced by a second term unlike (2.1):

$$\mathbf{x}_\mathrm{E} = \mathbf{f}(\mathbf{x}_\mathrm{b}, \mathbf{q}) \tag{2.7}$$

For example, Z. Vafa and S. Dubowsky developed the *Virtual Manipulator* approach to extend the fixed-base analysis tools to space robot. A fictitious fixed-based manipulator is derived based on the inertial properties of the whole system. Its anchorage point is chosen as the global CoM, which is supposed to stay fixed in the inertial frame with no external efforts, and its effector is located and oriented as the space robot one. In addition, the first joint is spherical and represents the inertial base attitude. An illustration is given in Figure 2.5a. Though well suited for kinematics and path planning, this method leads to complex dynamic equations and does not consider an actuated base.

The *Barycentric Vector* approach, illustrated in Figure 2.5b, is introduced by writing for each segment the inertial CoM position w.r.t. the global CoM (Umetani and Yoshida, 1989b; Papadopoulos and Dubowsky, 1991a). The computation of the kinetic energy turns out to be quite intricate, but, thanks to the inertial derivation, the global linear and angular motion are completely decoupled from the joint dynamics. In order to improve the computational efficiency, the *Direct Path* approach describes the position of any point along the spacecraft

w.r.t. the base coordinates instead of the global CoM (Moosavian and Papadopoulos, 2004). When written for the segments CoM, the kinetic energy and the dynamic equations are expressed in a compact form more suited for simulation. By contrast, these equations govern the base dynamics instead of the global position and attitude, and a post-treatment is thus necessary to extract these global quantities from the simulation data.

Kinetics When the kinematic relations are derived w.r.t. time, the Jacobian matrix is extended with the base motion. The end-effector velocities are written in a general way as:

$$\mathbf{t}_{\mathrm{E}} = \mathbf{J}_{\mathrm{b}}\,\mathbf{t}_{\mathrm{b}} + \mathbf{J}_{\mathrm{m}}(\mathbf{q})\dot{\mathbf{q}} \tag{2.8}$$

where $\mathbf{t}_{\mathrm{b}}$ and $\mathbf{t}_{\mathrm{E}}$ are the twists of, respectively, the base and the end-effector. The global Jacobian matrix is split into a base term, with $\mathbf{J}_{\mathrm{b}}$, and a manipulator one, with $\mathbf{J}_m$. Considering the free-floating case, whose base is not actively controlled, the momentum conservation brings an additional relation when the system is originally at rest:

$$\mathcal{H} = \mathbf{D}_{\mathrm{bb}}(\mathbf{q})\,\mathbf{t}_{\mathrm{b}} + \mathbf{D}_{\mathrm{bm}}(\mathbf{q})\,\dot{\mathbf{q}} = 0 \tag{2.9}$$

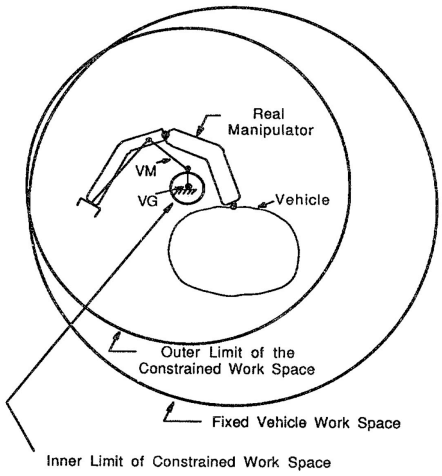
The inertia matrices of the base and the manipulator are denoted by $\mathbf{D}_{\mathrm{bb}}$ and $\mathbf{D}_{\mathrm{bm}}$. Combining both relations, it leads to the *Generalized Jacobian Matrix (GJM)* introduced for the first time by Y. Umetani and K. Yoshida in (Umetani and Yoshida, 1989b). It reads:

$$\mathbf{t}_{\mathrm{E}} = \mathbf{J}^{*}(\mathbf{q})\,\dot{\mathbf{q}} \;\text{ with }\; \mathbf{J}^{*} = \mathbf{J}_{\mathrm{m}} - \mathbf{J}_{\mathrm{b}}\,\mathbf{D}_{\mathrm{bb}}^{\#}\,\mathbf{D}_{\mathrm{bm}} \tag{2.10}$$

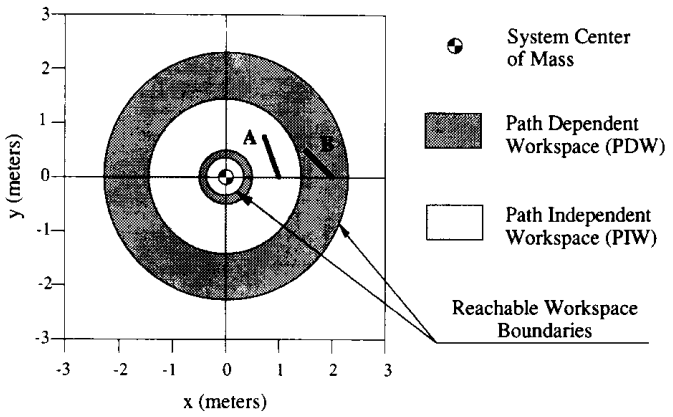
where the subscript # denotes the pseudo-inverse of a matrix.

Similar to the fixed-base case, inverse dynamics scheme using this new Jacobian matrix are introduced in (Caccavale and Siciliano, 2001). Nevertheless, since it now contains inertial terms through the $\mathbf{D}_{\mathrm{bb}}$ and $\mathbf{D}_{\mathrm{bm}}$ matrices, it does depend on the inertias and the masses of the manipulator and its base. Therefore, the singularities are not solely function of the geometric parameters and the manipulator configuration, but also of the inertial parameters. They are called *dynamic singularities* and were introduced by E. Papadopoulos and S. Dubowsky in (Papadopoulos and Dubowsky, 1989).

The space robot workspace is still partitioned according to these singularities. When the base is actively controlled, it is similar to the fixed-base case, but Z. Vafa shows how it is reduced drastically for a free-floating system (Vafa and Dubowsky, 1987). The difference is illustrated in Figure 2.6a. The workspace shrinks even further when the robot operates in a space free of



(a) Source: (Vafa and Dubowsky, 1987)

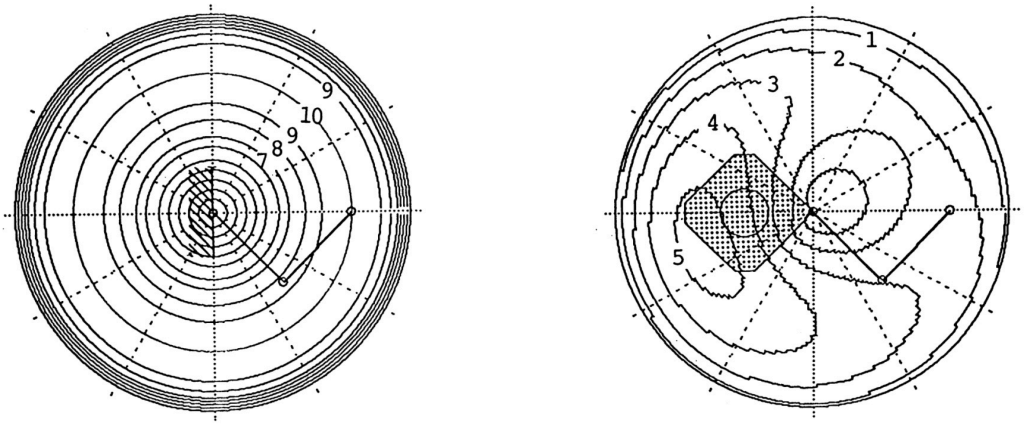


(b) Source: (Papadopoulos and Dubowsky, 1991a)

Figure 2.6: Workspaces analysis for free-flying/free-floating space robots. (a) Reachable workspaces derived from the Virtual Manipulator; (b) Cartesian space partitioning according to the dynamic singularities.

dynamic singularities. An example of space partitioning depending on these singularities is given in (Papadopoulos and Dubowsky, 1991a) and illustrated in Figure 2.6b. The two areas are denoted the *Path-Dependent Workspace* (PDW), also called the *Guaranteed Workspace* in (Yoshida and Umetani, 1990), and the *Path-Independent Workspace* (PIW). These names refer to the fact that, in the presence of singularities, alternative paths can be found to reach an inertial effector position while avoiding them. This crucial idea of dependency upon the path followed is due to the nonholonomic relation of the angular momentum conservation in (2.9) (Nakamura and Mukherjee, 1989, 1993). It means that a space manipulator can reach a given inertial position with different configurations, according to its base location and attitude. For example, the singularities can be avoided by changing the initial base attitude, prior to the manipulator motion (Nanos and Papadopoulos, 2012).

Lastly, a manipulability map is proposed in (Umetani and Yoshida, 2001) to measure the motion capabilities of a space manipulator. Around a given position, the effector can only move in a limited area before reaching a singularity, and the index of manipulability gives an indication on their proximity. It may be used to plan the manipulator task in areas maximizing its manipulability and avoiding the closest singularities. This analysis is led relatively to the base frame, and is presented for a planar case in (Umetani and Yoshida, 2001). On that example, an inverse kinematics convention is chosen to compute the arm configuration corresponding to an inertial position, either the “elbow up” or the “elbow



(a) Source: (Umetani and Yoshida, 2001) (b) Source: (Umetani and Yoshida, 2001)

Figure 2.7: Manipulability comparison between fixed-base and space robots. (a) Manipulability measure for a fixed-base robot with 2 DoF; (b) Manipulability measure for the same robot with a moving base.

down". It is clearly seen in Figure 2.7 how the manipulability index is drastically reduced when the base is moving freely.

Dynamics Concerning the dynamics of a space robot, the additional effort from the fixed-base algorithm is to model the free rigid motion of the base. Indeed, this latter is often considered as a rigid body on which various appendages are fixed, such as solar panels, antennas or robotic arms. Since the manipulator motion is usually planned for a short period of time, the orbital mechanics can be neglected and the system is considered to be in weightlessness[3] for the capture of a debris (Umetani and Yoshida, 1989b). The rigid space robot dynamics is derived using the Lagrangian approach and the barycentric vectors in (Papadopoulos and Dubowsky, 1991a,b), and the case of a multi-arm system is treated in (Papadopoulos and Moosavian, 1994). A compact model is presented and extended to humanoid robots in (Nenchev, 2013). When flexible appendages are embedded, the Lagrangian approach is still the most convenient one to model the whole system. The dynamics of a spacecraft with flexible solar panels and rigid arms is derived in (Ebrahimi and Moosavian, 2007), and a generic model of a flexible base is proposed in (Nenchev et al., 1999). The Jourdain's principle, equivalent to the Kane formalism, may also be used for these systems (Xu et al., 2014).

As mentioned above, there is some recurrent problems to model the rigid dynamics. Especially for the attitude motion, the Lagrangian equations become more intricate because the

[3]For longer robotic motion, the complete model with the gravity effects are investigated in (Wang et al., 2006; Guariniello et al., 2011).

kinetic energy of a rigid body is given by $T_{\text{rot}} = \frac{1}{2}\boldsymbol{\omega}^\top \mathbf{I}\,\boldsymbol{\omega}$, with $\mathbf{I}$ its inertia matrix and $\boldsymbol{\omega}$ its angular rate. This latter is not the immediate time derivative of any set of generalized coordinates (Hughes, 1986). For example, its relation with the Euler angles $\boldsymbol{\Psi}$ as generalized coordinates is given by a nonlinear equation: $\boldsymbol{\omega} = R_{\boldsymbol{\Psi}}\,\dot{\boldsymbol{\Psi}}$, with $R_{\boldsymbol{\Psi}}$ a transformation matrix described in (Craig, 1989). To overcome this limitation, *quasi-Lagrangian equations* are introduced by Meirovitch (Meirovitch, 1989) and Hughes (Hughes, 1986) to develop the *hybrid state equations*. The flexible dynamics is still derived with the classic Lagrangian formalism, but the rigid dynamics is based on the compact equations of Newton-Euler.

The derivation of linearized models is also crucial for the control of space robots. A dedicated tool is presented in (Alazard et al., 2008) to model any rigid tree-type system freely floating in space. In addition, any number of single flexible bodies can be attached to the base, in order to build more complex synthesis models. The solution, called *Satellite Dynamics Toolbox (SDT)*, is available online[4] and the underlying theory is given in (Alazard et al., 2008; Tantawi, 2007).

Simulation Regarding the simulation of space robot systems, the duality between Lagrangian and Newtonian approach still holds. On the one hand, the space robot dynamics are derived symbolically by augmenting the Lagrangian of the robotic arm with the base energy in (Moosavian and Papadopoulos, 2004). The model is computed once for all, and it allows for fast simulations, but it is only numerically tractable for a few DoF. Indeed, symbolic results become too large to store for more involved systems. On the other hand, C. R. Carignan considers the base dynamics by extending the recursive Newton-Euler algorithm of fixed-base robotics (Carignan and Akin, 2000). The efforts resulting from the manipulator motion are coupled to the base as external disturbances, while the segments kinetics is dictated by the base one.

A more general framework is provided by the DeNOC algorithms generalized for any tree-type system in (Shah et al., 2012a). The example of a humanoid robot is developed thoroughly and an efficient numerical algorithm is detailed. A toolbox dedicated to rigid multi-body systems is proposed with the *ReDySim* software[5], standing for *Recursive Dynamics Simulator* (Shah et al., 2012b). The modeling and simulation of legged and floating-base robots are covered. The present thesis extends these algorithms to include the dynamics of any flexible body inside the tree-type system.

[4] http://personnel.isae.fr/daniel-alazard/matlab-packages/satellite-dynamics-toolbox.html
[5] http://redysim.weebly.com/

2.2.3 Control strategies

The automatic maneuvers required to perform a rendezvous between two spacecrafts is covered in reference books like (Fehse, 2008). The Clohessy-Wiltshire equations may be used to describe the relative dynamics and then to design control laws and trajectory planning (Ankersen, 2011). This thesis rather focuses on the last steps of the capture, when the two spacecrafts can be considered floating freely in space, and when the target is within reach of the end-effector, or close to it.

Under these conditions, the control of a space robot may be chosen among two main avenues: the *free-flying* or the *free-floating* strategies. The first one consists of controlling both the base spacecraft and the robotic arm, while the second one is leaving the base free to react to any disturbances produced by the arm motion. This section goes through both strategies and presents the potential refinements.

Full Free-flying The free-flying mode is preferred when the base attitude has to be maintained during the arm motion. This is a strong requirement if the space robot is tele-operated from the ground, and if antennas need to be accurately pointed. As mentioned in (Dubowsky and Papadopoulos, 1993), any controller for the fixed-base robots may be applied if the base is precisely controlled and if the dynamic singularities are avoided. This is only realistic if powerful on-board actuators are used, like thrusters (Dubowsky et al., 1989; Dubowsky and Torres, 1991) or CMGs (Carpenter and Peck, 2009). When splitting the global controller between the base and the arm ones, a feedforward approach is presented in (Oda, 1994) to compensate for the arm disturbances on the base. The computational burden is lowered by approximating the wrist of the arm as a point mass in (Oda, 1996). On the contrary, the feedforward approach may provide the base efforts to the arm controller, in order to control the effector position and attitude by relative measurements, such as an on-board camera (Alexander and Cannon, 1989). In this work, a *resolved-acceleration* control is designed based on the explicit dependency of the effector acceleration with the control efforts of the base, extending thus the Operational Space Formulation (Khatib, 1987).

This is in sharp contrast with the "coordinated" control, when the base and the end-effector are managed in a cooperative fashion. For example, a coordinated controller is designed using the Transposed Jacobian scheme in (Papadopoulos and Dubowsky, 1991b), where an extended Jacobian matrix is built to describe the base and the effector motion at the same time. In addition, it is shown how the dynamic singularities can destabilize the controller when the Jacobian inverse is used to control the effector in the cartesian space. It is important to recall that the reaction wheels are too weak to compensate entirely for the disturbances

when the arm is moving fast, such that a coordinated control is necessary to perform agile motion. One potential solution is to use a redundant arm to plan reactionless trajectories (Spofford, 1991). A coordinated control is designed to reduce as much as possible the disturbances on the base, so they can be handled by the reaction wheels only.

Partial Free-flying A partial control of the base is also possible when the base position is not critical while its pointing accuracy must be maintained. This strategy, consisting in actively controlling the attitude while the position is left free, is usually denoted the "partial free-flying" control. The space robot kinematics is modified to account for the free translational motion during the end-effector path planning in (Longman et al., 1987). No control is thus necessary in translation, but the use of reaction wheels or thrusters is mandatory to manage the angular motion. An adaptive approach is proposed in (Walker and Wee, 1991) to cope with the uncertainties on the system inertial parameters.

Free-floating When fuel must be saved, the *free-floating* mode is preferred: the base is let free to translate and rotate when the arm moves, and the momentum conservation introduces the nonholonomic relation in (2.9). Based on the *Reaction Null-Space (RNS)* approach, D. N. Nenchev proposes a feedforward scheme combined with a reactionless path planner (Nenchev, 2013). A computed-torque controller is designed to cancel any disturbance, provided that the manipulator is sufficiently redundant (i.e., it has more degrees-of-freedom than the end-effector constraints). The same kind of controller is reduced to lower the on-line computations in (Koningstein and Cannon, 1991). By neglecting the base accelerations into the arm equations, the dynamics are reduced for the control, and the resulting errors are investigated on the real system. It is experimentally shown that this simplifying hypothesis is legitimate, when compared to the uncertainties on the masses and inertias of either the robot or the payload.

Jacobian-based control are also exploited for space robots using the *Generalized Jacobian Matrix (GJM)* presented above. An inverse-Jacobian scheme with a *resolved motion rate control* is developed and validated experimentally in (Umetani and Yoshida, 1989a,b). An extension of these works with a resolved-acceleration scheme is proposed in (Nenchev et al., 1999). In addition, the controller uses the RNS formulation to suppress the vibrations coming from a flexible base. Since this method is very sensitive to singularities, the transpose-Jacobian scheme is often preferred. An extension for space robot is presented in (Papadopoulos and Dubowsky, 1991a). With the additional hypothesis of a massive base, the generalized Jacobian is approximated by the fixed-base one in (Masutani et al., 1989). The resulting error is also investigated to assess the validity of this approach depending on the desired accuracy

for the effector positioning. To improve the control performances and to reduce the gain values, the *Modified Transpose Jacobian* is developed in (Moosavian and Papadopoulos, 1997, 2007b). Finding its roots in the feedback linearization technique, the proposed controller is based on a transpose-Jacobian form, and is augmented by a term depending on the previous commands to dampen the effect of nonlinear terms.

Adaptive approaches are also widely applied for the control of space robots. In order to preserve a reactionless motion of the arm, a scheme updates its inertial parameters when a target is grasped by the effector in (Nguyen-Huynh and Sharf, 2011). A main concern for adaptive control is the linear parametrization w.r.t. the estimated parameters. As shown in (Xu et al., 1994), the model of a space robot can be linearly parametrized by the dynamical parameters of the base, when it is expressed in the local frame instead of the inertial one. An extension is proposed by using neuronal networks to improve the effector accuracy and the energy consumption in (Vance and Sanner, 1996).

The mass distribution of a free-floating multi-body system is also investigated to give some design advices on the manipulator size and shape in (Agrawal et al., 2009). The main goal is to obtain a "flat system" because advantageous control and planning concepts follow, thus avoiding the use of computationally demanding techniques like nonlinear programming.

As a conclusion, both free-flying and free-floating strategies are usually brought together in a mission scenario. A comprehensive example is investigated for the whole capture scenario of a tumbling debris in (Yoshida et al., 2006). The *Bias Momentum Approach* is used first to synchronize the chaser with the tumbling target (Dimitrov, 2005), then the *Impedance Control* avoids to push it away during the contact phase (Yoshida and Nakanishi, 2003), and finally, the *Distributed Momentum Control* cancels the global angular momentum during the post-capture.

The previous sections dealt with the control strategies available for space robots. Once the controller structure is fixed, its tuning parameters are adjusted to ensure the desired performances and to meet the robustness requirements. In addition, a reference trajectory must be computed to automatically perform the capture of a tumbling object. The next sections cover, firstly, the path planning of a space robot depending on the control strategy, and secondly, the analysis and synthesis tools available to tune the control laws.

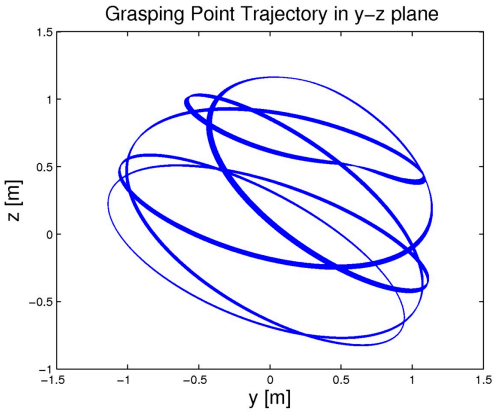
2.3 Guidance

Once the control strategy is fixed, the end-effector of the space robot can follow any reference input that lies into its workspace. This section focuses on the trajectory generation, and more specifically on the paths to safely capture a tumbling debris. The available strategies are still developed depending on the free-flying or free-floating control scheme of the space robot. In the sequel, the spacecraft to capture is called the "debris", while the specific point to grasp on it is referred as the "target".

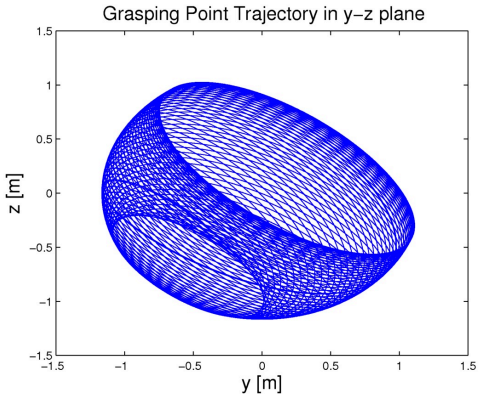
2.3.1 Target trajectory

First of all, the trajectory of the target point must be known precisely, whether by a sensor measure or by a model-based estimation. Performed during the ETS-VII mission, a vision-based servoing was developed to recognize visual markers on the serviced spacecraft (Inaba et al., 2003). In others experiments, the algorithm robustness w.r.t the harsh space environment and the illumination conditions is improved by using either a multi-camera configuration (Obermark et al., 2007), or by propagating the target dynamics (Lampariello, 2013). Some simplifying hypothesis can be made to alleviate the path planner complexity. By considering the target as a point mass, the dynamic singularities are avoided by following a straight line in the *Guaranteed Workspace* (Yoshida and Umetani, 1990). A circular target trajectory may also be used to approximate a flat-spin motion of the debris (Piersigilli et al., 2010; Nguyen-Huynh and Sharf, 2011). A more generic trajectory is computed by propagating the rigid-body tumbling motion with the dynamic parameters of the debris known a priori (Lampariello, 2013). Two examples from this reference are given in Figure 2.8, for two different initial conditions on the angular rate of the debris.

From a more practical point of view, the target dynamics will be estimated and improved on-orbit by measurement data from a camera. By merging the visual data and the estimation of the debris dynamics, the *Kalman Filter (KF)* provides an optimal estimator for the target trajectory (Nagamatsu et al., 1996; Aghili et al., 2010). This approach was even validated on the robotic test bench of the Canadian space agency (Rekleitis et al., 2007). Advanced concepts push further the *Extended Kalman Filter (EKF)* to focus on the estimation of the dynamics of a tumbling airship in weightlessness (Persson, 2015). When the debris is grasped, an identification of its inertial parameters, such as mass, moments and products of inertia, may be performed by using the gravity gradient effect in (Abiko and Yoshida, 2001). This identification refines the dynamic model of the debris and allows to update the controller accordingly to improve the performances.



(a) Source: (Lampariello, 2013)



(b) Source: (Lampariello, 2013)

Figure 2.8: Two trajectories of the target for different initial tumbling rate of the debris.

2.3.2 Path planning

The debris dynamics estimation provides the target trajectory that has to be intercepted for capture. When considering the space robot controller design, different approaches may be considered depending on the free-flying/free-floating configuration, and whether the primary goal is to minimize the base disturbance or to ensure a soft capture. In the following, the first section describes in more details the path planning methods to intercept the target trajectory, while the second one focuses on the criteria that can be optimized when computing the capture trajectory.

2.3.2.1 Target interception

The capture can be realized by matching only the target position, or both its position and speed. Without the speed continuity at capture, an impact must occur but could be minimized by using the percussion point theory in (Papadopoulos and Paraskevas, 2006). To avoid pushing away the debris from the robot, an impedance control is also presented in (Yoshida and Nakanishi, 2003). On the other hand, if speed continuity is ensured at capture, no impact is expected in theory. A closed-form solution is presented in (Aghili, 2009a), by using the Pontryagin's minimum principle (Athans and Falb, 1966; Bryson and Ho, 1975). The goal is to minimize a cost function and to match the target position and speed at the instant of capture. In the case of noisy measurements and parametric uncertainties on the debris inertia matrix, a complete scheme is proposed in (Aghili, 2012) to estimate and predict its trajectory using a Kalman filter, and thus ensuring a more robust capture.

Nevertheless, no study consider the acceleration continuity when designing the capture tra-

jectory. By switching in a smoother way to the free target trajectory, this last requirement avoids any discontinuity in the commanded torques at the instant of capture. Doing this, the grapple fixture can be tracked during a short period, and then the capture mechanism will be closed in a safer way. The corresponding path planner is one of the contributions of this thesis.

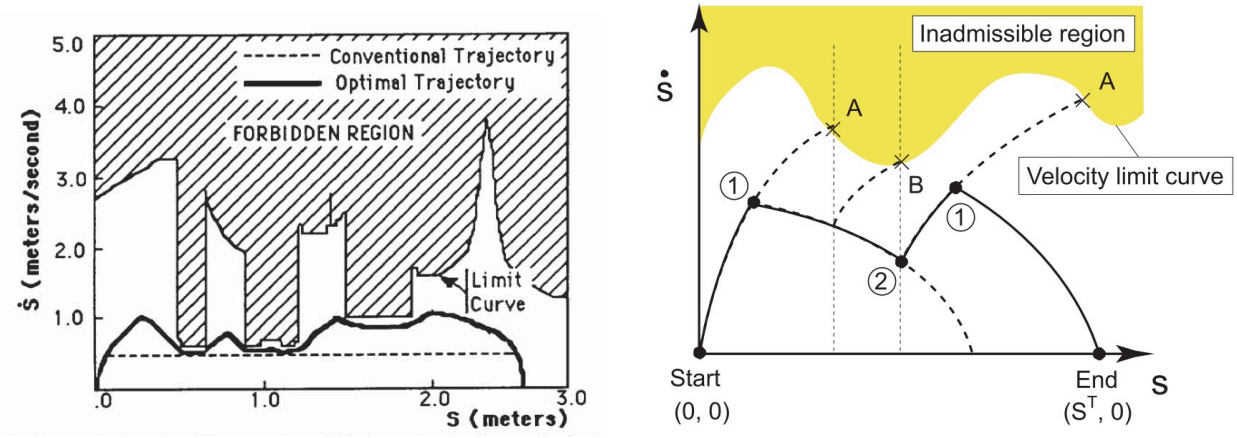
2.3.2.2 Optimization criteria

Many criteria can be considered to design the capture trajectories. Among the most important ones, one may need to reduce the fuel consumption of the base, or to avoid the obstacles during the manipulator motion. An overview of the criteria proposed in the literature is given in this section.

Actuator efforts and saturations When the base is actively controlled, the *Limit Curve* concept is introduced in (Dubowsky et al., 1989) to compute the time optimal trajectory without saturating the base thrusters. The speed is increased along the trajectory, as long as the arm configuration does not produce disturbances higher than the actuators capabilities, as illustrated in Figure 2.9. A similar concept is studied with reaction wheels in (Oki et al., 2008). A special focus is put on the choice of switching points where the arm must be slowed down or where it can be accelerated, depending on whether the trajectory was, respectively, reaching, or going away, from the limit curve. Instead of specifying limitations on the actuators, a set of requirements might also be imposed on the maximum angular speed and drift of the base (Oda, 1997).

From the point of view of spacecraft rendezvous, a simultaneous minimization on time and fuel is developed in (Ma et al., 2007). The required thrust profile is computed to synchronize the chaser linear and angular motion with a tumbling object in (Boyarko et al., 2011). The 6DoF rendezvous is developed in a circular orbit, and it is built on numerical methods to solve the Pontryagin's minimum principle. Unfortunately, it remains too cumbersome to consider a real-time application. Nevertheless, the use of such a method would allow to move the space robot such that the target trajectory stays into the manipulator's workspace.

Manipulator disturbances and base attitude When the AOCS efforts must be reduced, the manipulator motion can be planned to minimize the disturbances produced on its base. For example, the singular values of the submatrix $\mathbf{D}_{\mathrm{bm}}$ introduced in (2.9) are used to draw the *Disturbance Map*, which indicates a direction of motion with minimal disturbances (Dubowsky and Torres, 1991). This tool was initially developed for a two-links manipulators, and is extended for any number of joints with the *Enhanced Disturbance Map* in (Torres and



(a) Source: (Dubowsky et al., 1989) (b) Source: (Oki et al., 2008)

Figure 2.9: *Limit curve* concept and switching point choices.

Dubowsky, 1992). It is further developed to encompass the flexible behavior of the base with the *Coupling Map*, in order to limit the base vibrations produced by the manipulator motion (Torres and Dubowsky, 1993). This particular matrix $\mathbf{D}_{\text{bm}}$ gives the participation of the manipulator to the global momentum, and it also couples the base and the manipulator states in the global dynamics. In addition, this is the foundation of the RNS approach, which provides effector trajectory with no reaction on the base (Nenchev, 2013). This approach is applied with an adaptive control scheme to maintain a reactionless motion while capturing a target (Nguyen-Huynh and Sharf, 2011). Another planning method is developed in (Yamada et al., 1995) to efficiently use the momentum stored in the reaction wheels to dampen the spacecraft angular drift and to avoid any nutation.

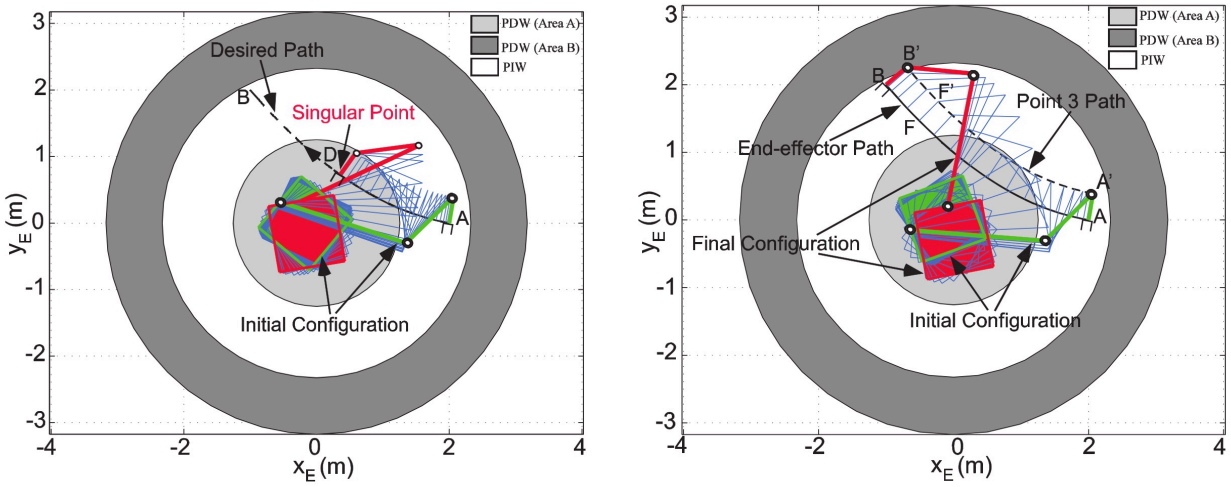
When the base is considered free-floating, the momentum conservation introduces a non-holonomic relation. Using this system property, the *Bi-Directional Approach* is proposed in (Nakamura and Mukherjee, 1991) to control simultaneously the base orientation and the manipulator's joints with the joint torques only. A performance index is even introduced, called the *Coupling Factor*, to measure the influence of the manipulator motion on the base, depending on its configuration (Xu, 1993). An alternative technique is proposed with the *Self-Correcting Motions* to modify the base attitude, not with the AOCS, but with the arm itself (Vafa and Dubowsky, 1987). This approach avoids any fuel consumption and relies only on the electric consumption of the joints. It consists in performing cyclic maneuvers of the arm to adjust the base attitude as needed.

The drawback of the previous method is to correct periodically the angular motion of the base

to recover a given pointing performance. Therefore, no large motions of the arm are possible, or the communication signal would be lost. A solution is brought by the *Fixed-Attitude Restricted* motion, using redundant manipulators (Nenchev et al., 1992). The path planning is based on a velocity reference taken in the kernel of the manipulator Jacobian Matrix, in order to maintain the base attitude. Similar to fixed-base robotics, a manipulability measure is also derived to help analyzing and designing the path-planner of space-based manipulators.

Obstacle avoidance Apart from minimizing the disturbances on the base and the related energy consumption, the path planning must also deal with the potential collisions with the obstacles surrounding the manipulator. As presented in (Lampariello, 2013), the trajectories can be parametrized as B-splines, whose variables are optimized to reach the target while avoiding the obstacles. The underlying optimization problem with the obstacle avoidance constraint is solved numerically using the single shooting method. The advantage of a numerical method is to account for all sort of constraints as the limits on the joint positions and speeds, or the avoidance of inner collisions between the segments themselves. A real-time implementation based on lookup-tables is proposed in (Lampariello, 2010), even though the resulting algorithm is time-consuming and requires a substantial computation power. In a more analytic fashion, the obstacles can be mapped into an intermediate space, where the space robot configurations are also projected in a one-to-one manner that prevent any singularity (Papadopoulos et al., 2002). Similar to the techniques of mobile robotics, the trajectories are then represented by polynomials, with a degree sufficiently high to avoid all the obstacles. Search algorithms are also developed around the concept of the *sampling-based* A^* algorithm in (Persson, 2015). It provides an efficient motion planner of the robotic arm, that can avoid obstacles and joint limits, and even optimizes a given cost function .

Dynamic singularities avoidance A final area of concern is the management of the dynamic singularities mentioned earlier. An inertial trajectory planning is proposed in (Nanos and Papadopoulos, 2012, 2015) to avoid the path-dependent singularities. The initial system configuration is carefully chosen, in order to reach the final inertial position of the effector without passing by a singularity as shown in Figure 2.10. It implies to re-orient the base and the arm before performing the maneuver. This approach is consistent with the arm's ability to re-orient the satellite by cyclic maneuvers, as it takes advantage of the nonholonomic nature of the system (Vafa and Dubowsky, 1987). The main drawbacks are the free-floating hypothesis and the rotation of the base, that prevent any ground communication during the motion.



(a) Source: (Nanos and Papadopoulos, 2015) (b) Source: (Nanos and Papadopoulos, 2015)

Figure 2.10: Singularity avoidance by modifying the initial configuration of the space robot; (a) Singularity coming up with a random initial configuration, (b) Avoidance of the singularity with a new initial configuration.

2.4 Controller Synthesis

In order to follow the previous capture trajectories, the space robot controller must be efficiently designed and tuned. In the light of the above comments, the control of such systems faces many challenges. Firstly, the gain synthesis is mainly based on linear models, obtained by linearizing their highly nonlinear dynamics. Secondly, the presence of lumped or distributed flexibility forces to perform extended analysis to assess the stability and performances of the closed-loop system. Finally, uncertainties occur at many levels and require to evaluate the robustness properties of the controller.

2.4.1 Synthesis methods

Once the controller structure is fixed, stability and performance are ensured by carefully tuning its gains. As highlighted for the rigid case by (Kawamura et al., 1988), a PD controller may achieve any trajectory tracking performance with a sufficiently high derivative action. Nevertheless, limited gain values are considered to avoid actuators saturation, while a trajectory tracking performance is still needed. To meet this requirements, the so-called *fixed-structure* H_∞synthesis has known a growing interest in the last years with the addition of controller constraints in the original optimization problem (Gahinet and Apkarian, 2011a). The classic H_∞synthesis yielded an optimal tuning which complies with internal

stability and frequency domain performances, but for a full order controller. As emphasized in (Apkarian and Noll, 2006), optimization techniques are necessary as soon as general constraints are specified, like controller order, decentralized control or PD-like structure. A nonsmooth optimization method has been introduced by P. Apkarian and D. Noll to overcome the standard H_∞limitations. It allows to specify the controller order and/or structure (e.g., decentralized, PD, delay, observer-based...), to impose frequency domain constraints for performance purpose (e.g., response time, bandwith, disturbance rejection...), and to perform multi-model synthesis (Gahinet and Apkarian, 2011b). This approach avoids the reduction of an optimal full order controller obtained through classic H_∞synthesis solvers. An efficient implementation is available since the 2012b release of the *Robust Control Toolbox* of the `Matlab`© software (Gahinet and Apkarian, 2011a; Apkarian, 2013). In this *fixed-structure* H_∞framework, the unmodeled dynamics or the disturbance rejection can be handled by using weighting functions on the acceleration sensitivity function, rather than on the position or velocity ones (Alazard, 2013).

Reminding that space robots are driven by space processors with limited capabilities, linear controllers are preferred for their simplicity and their low computational load. In this case, a gain-scheduling approach appear necessary to handle the highly nonlinear system behavior and to design a controller valid on the widest workspace (Rugh and Shamma, 2000; Leith and Leithead, 2000). Indeed, when a joint of the arm is moving, great changes appear in the dynamic model and some terms can even change sign. An example for a flexible fixed-base robot with two DoF is proposed in (Apkarian and Adams, 1998). The controller is scheduled w.r.t. the arm configuration. Unlike random uncertainties, a specific structure for the uncertainty block can be derived, and more robust and efficient controllers are synthesized with the proposed technique. Experimental results are also presented for a flexible manipulator with a controller scheduled on the whole workspace (Carusone et al., 1993). Synthesis are performed on many linearized models of the rigid dynamics, and the scheduled controller is obtained by a linear interpolation among these gains. The main advantage is to provide a controller valid for any desired trajectory, but at the expense of a large memory storage.

The scheduling variable could also be the tilt angle of a solar panel, the size of which also has a strong impact on the dynamic model and on the spacecraft flexible modes (Nagashio et al., 2011). The next section addresses more specifically the control of such a flexible system.

2.4.2 Flexible control issues

The control of a flexible structure still remains a challenging task, mainly because of **non-collocated feedback** schemes and due to the presence of flexible modes in the vicinity of the desired bandwidth. Derived from a classic PD controller, a phase control is developed in (Alazard and Chrétien, 1993) to overcome the limitations of both collocated or non-collocated control alone. Indeed, the collocated feedback allows to dampen the in-joint flexibility with reasonable margins, but they shrink when the controller is sampled. On the other hand, a non-collocated scheme feeding back the flexible deflection does not succeed in damping enough the rotor mode.

Considering flexible links, the system exhibits a **non-minimum phase property** that is a strong performance limitation (Skogestad and Postlethwaite, 2005). The singular perturbation technique can be used to design a composite controller that stabilizes the fast flexible dynamics, and ensures the minimum phase property of the reduced low dynamics (Hashtrudi Zaad and Khorasani, 1996). A variation is proposed in (Masoudi and Mahzoon, 2011), where the low dynamics corresponding to the rigid motion are controlled in open loop, while the fast flexible dynamics are suppressed by a feedback control tuned by an LQR approach.

Attention must also be paid to the **spill-over effect**, i.e., the potential instability of the closed-loop system when high-frequency flexible modes are neglected in the synthesis. Indeed, a high controller bandwidth is necessary to ensure the performances, but it may excite the unmodeled flexible modes and could result in an unstable closed loop. In the scope of space robots, a matrix is extracted from the dynamic equations, and the sign of its terms indicates if the spillover is avoided or not (Gasbarri and Pisculli, 2015). A comparison is also proposed to highlight the benefits of using the H_∞ synthesis over the classic LQR one in (Grewal and Modit, 1995). Against the unmodeled dynamics like high-frequency flexibilities, the H_∞ synthesis performed better thanks to a strong roll-off specification on the system output. To go further on practical aspects, an optimal configuration for the number and placement of piezoelectric devices is developed to actively control and suppress the vibrations on a fully flexible space robot in (Sabatini et al., 2012).

As mentioned earlier, the flexible behavior can be represented by many different techniques and the modes shapes or frequencies may vary greatly depending on the boundary conditions. This leads to new uncertainties. In addition to the unstructured uncertainties due to unmodeled high-frequency modes, the flexible modes introduce a structured uncertainty because they shift when an element of a multi-body system moves. Considering the example

of a spacecraft equipped with rotating solar panels, a gain-scheduled controller is designed to cope with both uncertainties and is validated by on-orbit experiments with the ETS-VIII mission (Nagashio et al., 2011). More insights are given in the next section about the robustness issue for manipulators and flexible systems.

2.4.3 Robustness analysis

From the robustness point of view, a rigid robotic system can be set under one of the many *Linear Parameter Varying (LPV)* forms available in the literature, as illustrated in (Löhning, 2010) for the *Linear Fractional Transformation (LFT)*. The joint angles are considered as bounded uncertain parameters. They parametrize the nonlinear model of a 3-DoF robot, including the mass matrix and the vector of Coriolis and centrifugal terms. This approach can be extended to more complex robots with a 6-DoF manipulator (Saupe and Pfifer, 2012). The first axis controller is designed considering that the load inertia seen by the motor is a varying parameter. To handle the model dependency on all the joint angles, the *parameter set mapping* is also introduced in (Hashemi et al., 2012). By reducing the number of parameters used in the LPV model, the controller synthesis is simplified and computational complexity is lowered.

The drawback of these methods is to consider the analytical expression of the mass matrix terms to compute their extreme values when the arm moves. A more general approach is obtained by first deriving the LFT form of a single rotation matrix w.r.t. its angle, as introduced in (Manceaux-Cumer and Chrétien, 2001). With this LFT form, the dynamics is readily parametrized by this rotation angle. The synthesis of a robust controller is then performed for a satellite with varying tilt angles of the solar arrays. A smaller LFT representation of the rotation matrix is also proposed in (Guy et al., 2014). This approach allows to derive the LFT model of complex spacecrafts with uncertainties on rigid and flexible quantities at the same time (Loquen et al., 2012). A controller is designed for a satellite with flexible solar panels and antenna. Based on the H_∞ and the μ frameworks, an analysis is performed to highlight the limitations of the robust performance when the uncertainties are becoming too important.

In addition, some key parameters might be considered as additional optimization variables in the design process. Called the *Co-Design* synthesis, this approach allows to simultaneously optimize the system parameters, such as lengths or masses, along with the gains (Alazard et al., 2013). It provides a powerful tool for a preliminary design of both the system itself and its controller. By optimizing these key parameters, it may lead to improved performances.

2.5 Testing and Validation

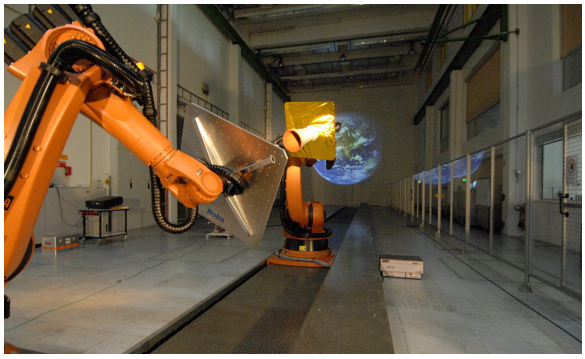
Finally, the closed-loop system and the path planner must be brought together to evaluate the full and safe capture of a debris. A first step is provided by the simulation tools described earlier and further developed in this thesis. But an experimental validation is necessary to assess the real performances, and raises the issue of reproducing the space conditions on Earth. In addition to the lighting conditions, the temperatures, and the space radiations, the weightlessness dynamics is one of the biggest challenges. In this section, the main avenues to test and validate space robotic systems are reviewed and classified from the cheapest to the most expensive.

Suspension systems According to (Menon et al., 2007), the suspension of the segments by long wires and springs allows to compensate for their weight. Low and small displacements are required to prevent the system from getting out of this artificial weightlessness condition. Indeed the cables do not remain exactly vertical when the manipulator moves and thus the gravity is not fully compensated in long-reach motion. The main advantage of this solution is obviously its modest cost.

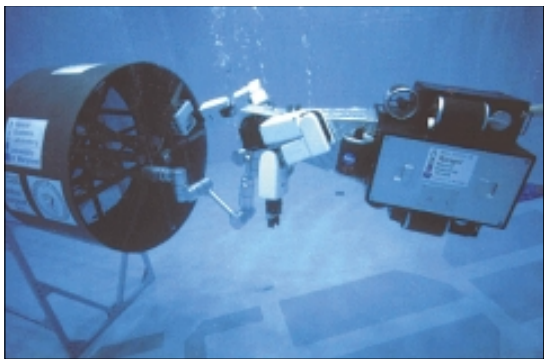
Air-bearing planar systems This system consists of a set of air-cushion pads located under each segment and under the base. The pressurized air cancels the ground friction, and fully compensates for the gravity. Therefore, the base reacts exactly as if it were in conditions of weightlessness. Moreover, long-time experiments can be led with no additional cost. The main drawback is the limitation to a planar motion. This setup is very popular though, because of its cost-benefits ratio. The first space robotics experiments were conducted with these systems (Alexander and Cannon, 1989), including for the ETS-VII mission (Umetani and Yoshida, 1989a). Many experiments are also based on air-bearings, by the Space Research Center in Poland (Rybus et al., 2013; Rybus and Seweryn, 2013), and by the University of Padova in Italy (Cocuzza et al., 2011), among others.

In addition, these test benches appear well-suited for the validation of flexible robots introduced in (Nenchev et al., 1999) and in (Alazard and Chrétien, 1992), which are planar as well.

Industrial robots Kinematics, trajectories and hardware devices can be validated with industrial robots. The German space agency DLR possesses this kind of test bench with the EPOS project illustrated in Figure 2.11a. By reproducing the lighting conditions with glaring and reflections, an embedded camera and its image processing algorithm are tested



(a) Source: (Boge et al., 2010)



(b) Source: (Carignan and Akin, 2000)

Figure 2.11: Test benches to validate space robotics on Earth; (a) EPOS test bench developed by the DLR; (b) Neutral Buoyancy Ranger from the University of Maryland.

and validated almost in real conditions (Boge et al., 2010). One robotic arm reproduces the rotating motion of a target with a mockup of the debris, while the second one is translating on a rail and stands for the chaser on approach. Since the control loop is closed by the visual feedback, this approach is called HIL.

A similar test bench is presented by the *Canadian Space Agency (CSA)* in (Rekleitis et al., 2007). For this one, both robots are static but the principle remains the same. One is moving like the target and the other one like the chaser. High level task were performed based on the *Cortex* control software, in order to demonstrate autonomous steps of a capture mission. It is also worth mentioning the Chinese test bench made up of two static industrial robots (Xu et al., 2007), and the neutral-buoyant airship developed at the McGill University (Persson, 2015). An exhaustive list of the similar setups is available in (Flores-Abad et al., 2014).

Under-water systems In order to improve the on-orbit dynamics simulation, under-water systems have been designed. They are used intensively for the astronauts training phases on Earth, and have recently been proved to be very similar to the weightlessness conditions (Macaluso et al., 2016). Thanks to the Archimedes' buoyancy principle, weightlessness can be reached for big systems provided that they are built to be waterproof and that they can fit into a pool. 3D motion can also be tested and long-time experiments are feasible. However, when an object is moving underwater, it experiences hydrodynamic effects and friction that do not occur in space. An advanced hydrodynamic study is provided in (Menon et al., 2007), and it shows that a spherical object actuated with hydro-jet nozzles can accurately reproduce the behavior of a spacecraft.

Using the *Neutral Buoyancy Vehicle (NBV)* from the University of Maryland, C. R. Carignan

performed underwater experiments (Carignan and Akin, 2000), as illustrated in Figure 2.11b. The base displacement is observed when a manipulator motion is performed with the AOCS turned on and off. It illustrates the necessity of the free-flying strategy over the free-floating one when the base has to be precisely pointed.

Parabolic flights When a plane performs parabolic trajectories, the gravity felt by the passengers can be augmented by the descent rate and the resulting centrifugal force. On the contrary, when it is rising the gravity can be canceled by this same centrifugal force. This method allows to reach micro-gravity conditions for almost 30 seconds, it can be repeated dozens of time on a single flight and can validate systems which fit inside the cabin. Therefore, only short-time experiments are possible in a limited workspace, and this solution still remains quite expensive. However, it reproduces quite faithfully the weightlessness conditions with only 5% of the terrestrial gravity, and it is also shown that mockups of space robots can be embedded (Menon et al., 2007). The French company *Novespace* provides this service with a A300 plane for a cumulative time of ten minutes in micro-gravity on one flight[6].

On-orbit experiments Eventually, the most expensive way of testing space systems, and the most faithful one, is to bring them up to space. As highlighted in the introduction of the thesis, many missions provide strong feedbacks on space robotic systems, such as the the experiments of the ETS-VII, of the OE, or of the *Canadarm*. A potential low-cost solution has been proposed in (Seweryn et al., 2011), by using the *Columbus* module of the ISS to let the astronauts perform the experiments. A similar project has been brought to life by the SPHERES spacecrafts of the *Massachusetts Institute of Technology (MIT)*. They were sent aboard the ISS to investigate the proximity maneuvers of a chaser with a target, to build a 3D map of the debris, and to perform a rendezvous while avoiding obstacles (Stoll et al., 2012).

[6]http://www.novespace.fr/en,home.html

CHAPTER 3 RESEARCH OBJECTIVES

The main objectives of this thesis are based on the previous literature review. By highlighting the advantages and limitations of the current methods used for the modeling, the control and the path planning of space robots, the expected contributions are presented. In the present work, algorithms are developed for the flexible modeling and the control synthesis of such robotic system to capture a tumbling debris.

The mission scenario is first described to highlight the limits of the project. Then, the objectives and their corresponding contributions are detailed, before introducing the global organization of the thesis.

3.1 Mission Scenario

This thesis aims at studying the modeling and the control of a space robot to capture a tumbling debris. Too many questions are surrounding such a challenging task, and some hypothesis are necessary to narrow the field of research. To this end, only the planning and the capture phases of the whole mission are studied. They correspond to the steps of fly-around and of mechanical interfacing, illustrated in the global scenario in Figures 1.7c and 1.7d. The project aims at modeling faithfully a space robot with lightweight flexible segments, in order to build a high-fidelity simulator and to derive a linearized model for the control synthesis. The validation of the designed controller is performed by the study case of a small debris tumbling at a rate of $4°/s$, and using a chaser similar to past or existing projects, like the ETS-VII or the DEOS. An illustration of the considered phase is given in Figure 3.1.

With these limitations on the mission scenario, the capture is assumed to last a small amount of time, compared to the period of the spacecraft trajectories around the Earth. Therefore, orbital mechanics will be neglected and the weightlessness hypothesis is used in the modeling. Nevertheless, the on-orbit disturbances are introduced as external efforts applied on the spacecraft supporting the manipulator.

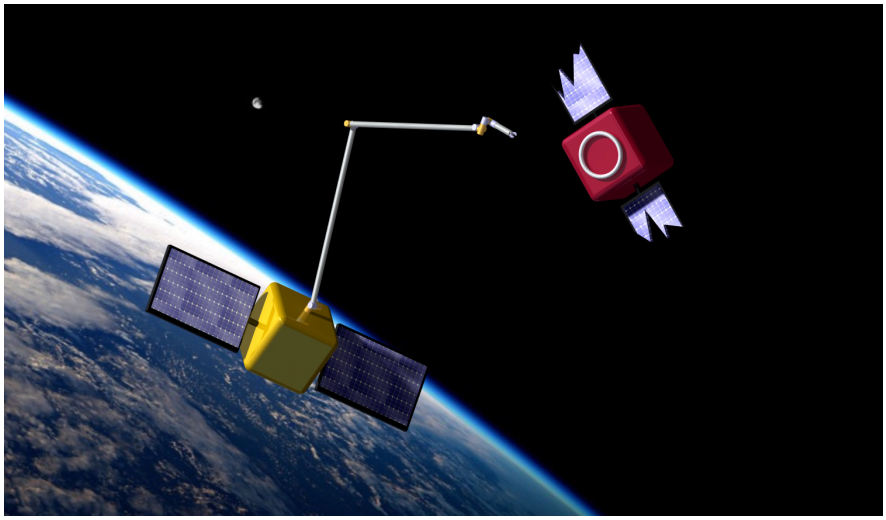


Figure 3.1: Illustration of the on-orbit capture of a debris by a space robot

The main features of the thesis may be summarized by:

- Multi-body dynamics in weightlessness;
- Chaser considered as a rigid moving base with chain-like flexible appendages;
- Manipulator described by chain-like structures, with flexible and rigid segments;
- Debris considered as a tumbling rigid body.

while the underlying hypotheses are listed below:

- Orbital mechanics neglected;
- In-joint flexibility of the manipulator neglected;
- Tree-type systems neglected;
- Mechanical closed-loops neglected;
- Actuators and sensors assumed to be perfect.

3.2 Thesis Contributions

With this clear description of the thesis limits, the objectives and the contributions are described in more details according to the four main avenues of the project, namely, the multi-body modeling, the path planning, the control synthesis and analysis, and the experimental validation.

Main contribution The main contribution of the thesis is proposed based on the extended literature review provided earlier, and on the hypotheses made above. It might be summarized as follows:

Original Contribution

Design of the control law of a space robot to perform the capture of a tumbling debris, and its validation by numerical simulations, and experiments with industrial manipulators.

Specific axes of development This main contribution is built on the next specific objectives. Each of them is formulated with the related hypothesis, and a short discussion describing the conditions considered to assess their fulfillment. The four specific objectives are denoted by **Oi**, and their hypothesis by *Hi*.

O1 - Derivation of the nonlinear flexible model of the chaser dynamics to build linearized synthesis models and a high-fidelity simulator.

H1 - The nonlinear flexible model is linearizable, and the simulation of the multi-body system is numerically tractable with a stable error on the energy drift during the simulations.

This hypothesis is validated if the nonlinear dynamics of the chaser can be linearized around an equilibrium point, as it is the case for most mechanical systems. Concerning the simulation, the energy drift must be monitored to check that the numerical errors are not increasing constantly during the computations. Otherwise, the results would be distorted and the numerical simulation will not be an efficient mean of validation. This objective will be validated on the complete chaser with a rigid base, two flexible solar panels, and a flexible manipulator as well.

O2 - Design of a coordinated control law and of a path planner for the spacecraft and its embedded manipulator, allowing to capture a tumbling debris from a few meters.

H2 - A separated control law on the spacecraft and on the robotic arm is able to track a capture trajectory within an accuracy of a few centimeters, in order to synchronize precisely the end-effector motion with the target point one.

The control law is designed for the specific case of the capture of space debris tumbling at a low rate. The base consumption and the end-effector positioning will be evaluated to assess the global performance of the law. A path planner must be developed to track the debris in a smooth way, avoiding any impact at the instant of capture. Two different controllers will be investigated on the manipulator, in order to simplify the law as much as possible and make it compatible with space processor capabilities. Their accuracy in simulation must reach an error of a few centimeters, as required in the previous space missions involving a capture.

O3 - Analysis of the controller robustness w.r.t. the important dynamic changes related to the manipulator motion.

H3 - A robustness analysis provides the extreme variations of the joint angles allowing to maintain the required performances on the linearized models of synthesis.

This hypothesis accounts for the parameter-varying structure of the linearized model of a manipulator. When the joint configuration is modified, then the linearized model changes completely and the controller validity must be checked on the new configuration. This problem will be overcome by specifying the maximal variations of the joint angles, in order to maintain the performances. Therefore, the domain of validity of the closed loop performances with one given controller can be derived in terms of joint angles variations around the nominal configuration. This study is based on the μ-analysis of the linear model of a robotic system, and must be performed for both the rigid and the flexible cases.

O4 - Validation of the control laws by performing a complete capture scenario on a robotic test bench.

H4 - Numerical simulations and industrial manipulators allow to faithfully reproduce the multi-body dynamics in weightlessness, in order to measure precisely the capture accuracy of a given controller.

The main problem for this objective is to reproduce the multi-body dynamics in weightlessness with terrestrial manipulators undergoing the gravity. The coupling of both the numerical and experimental simulations must lead to the validation of the capture with *Hardware-In-the-Loop (HIL)* components like the chaser camera tracking the debris position and attitude. The resulting error at capture remains the criterion for the mission success.

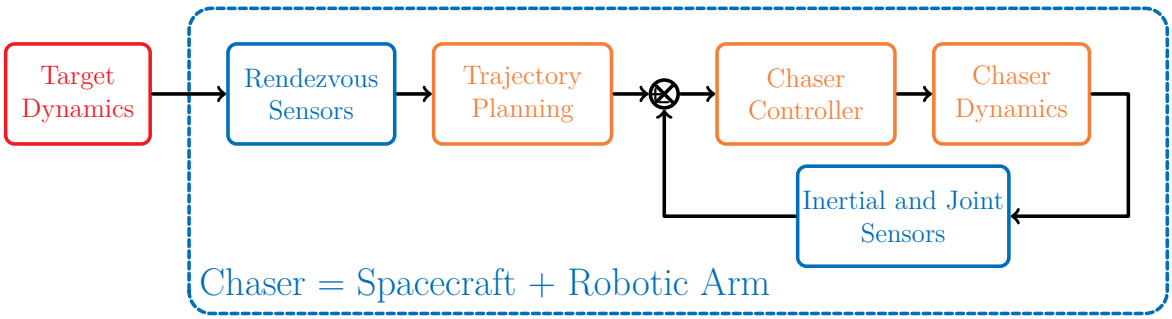


Figure 3.2: Global control architecture of the system and topics of the thesis in the orange blocks

3.3 Thesis Content

The thesis is organized into three main parts: the modeling, the control, and the validation. Various chapters are included in each part as described below. The main contents and results are in these parts, while the detailed computations, proofs and physical data used in each chapter are gathered in the appendices. The global architecture of the project is illustrated in Figure 3.2, where the specific topics of the thesis are emphasized by orange blocks.

Part 1: Modeling The first part deals with the modeling of a space robot using the DeNOC framework. The notations and the main steps are first introduced for rigid manipulators, before being extended to account for the segment flexibility. The resulting flexible dynamics of a manipulator is derived under the same form as for the rigid ones, using common notations for the matrices involved in their kinematics, kinetics and dynamics. The corresponding algorithms to compute the forward and inverse dynamics in simulation are extended to include the motion of the base. This model describes the space robot as a rigid base with various appendages, that can be either rigid or flexible, either fixed or actuated. The global model is derived under the weightlessness hypothesis, mainly considering the efforts in orbit as external disturbances.

The main contribution on this topic is the derivation of flexible dynamics using the DeNOC approach. It allows to clearly separate the dynamical terms which depends on the flexible coordinates, from the ones that remain constant. These corrective terms, that need to be updated during the simulation, are neglected to build approximate models, numerically more efficient and still accurate. A trade-off between the computation time reduction and the resulting accuracy is proposed in order to recommend the best simulation model.

Part 2: Path planning and control The second part aims at designing a control law for this complex system, by coordinating two separate controllers on the base and on the manipulator. The main goal being to follow a capture trajectory, a path planner is first developed to ensure that the reference followed by the space robot will result in a safe and smooth capture of the debris. Secondly, the controller syntheses are performed using the H_∞ framework with a requirement of disturbance rejection on the acceleration sensitivity function. This feature allows to better take care of the unmodeled dynamics. The global controller is split between two independant ones for the base and the arm. For this latter, two different control architectures are synthesized and compared: a centralized structure, that couples the measurements of all joints, and a decentralized one, that performs a "blind" control at each joint. These two cases allow to compare the trajectory tracking performances for both controllers in order to advise the best compromise between the performance and computational standpoints.

The robustness is also studied to assess the domain of validity of these controllers when the arm configuration is modified. To do this, the LFT modeling of a single rotation matrix is derived under the minimal form, and then applied recursively in the dynamic algorithms. The resulting models of any multi-body system is parameterized by the joint angles and dedicated tools are used to study the robust stability and performance by μ-analysis. This is the first time that a manipulator mass matrix is derived under the LFT form, and this work provides a starting point to validate the robust behavior of robotic controllers.

Part 3: Validation Eventually, the last part is dedicated to the validation of the control law and of its path planner. The numerical validation is first investigated by testing the control architecture designed earlier. The results are discussed according to the different strategies used to control and guide the end-effector and the base. Then, the experimental results obtained on the robotic test bench are presented. A global overview is given of the *Thales Alenia Space (TAS)* facility. The capabilities and limitations of this bench are discussed in the scope of space robotics, before presenting the results of a capture obtained by a mixed numerical and physical simulation. The main contribution of this work is the successful simulation of a closed-loop system integrating *Hardware-In-the-Loop (HIL)* components, with a real camera and its dedicated image processing software.

CHAPTER 4 RIGID FIXED-BASE ROBOT MODELING

The dynamic equations are first developed for a rigid body, in order to express the classic Newton and Euler laws of motion using the *Decoupled Natural Orthogonal Complement (DeNOC)* approach. This latter is especially useful to write kinematics, kinetics and dynamics in a compact form, mainly based on the general screw theory (Featherstone, 2008). In addition, the quasi-coordinates are introduced in this section for the simple rigid case, before extending them to the more intricate flexible case.

4.1 Model of a Single Rigid Segment

The model of a single rigid segment constrained at both ends is developed by applying the quasi-Lagrangian equations ((Hughes, 1986), p.60). The kinetic energy of the whole body is obtained based on the twist describing the body motion, while its derivation w.r.t. the system states yields the dynamic equations.

4.1.1 Kinematics

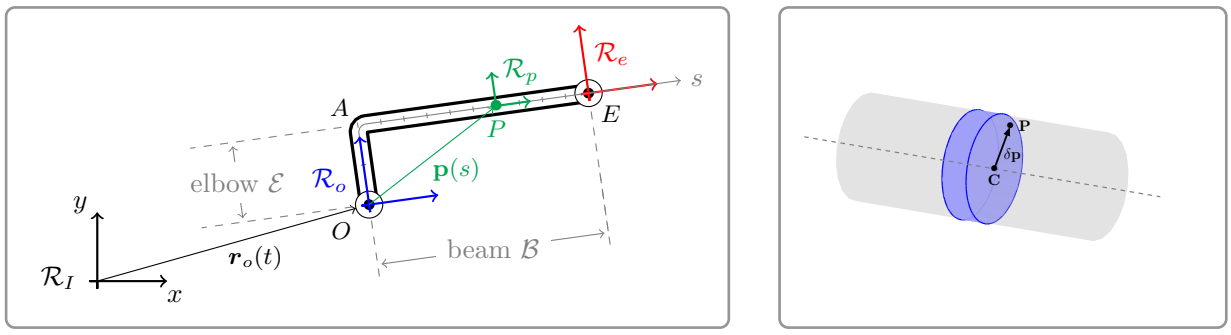
Three different frames are introduced in Figure 4.1: $\mathcal{R}_{\mathrm{I}}$ is the inertial frame, $\mathcal{R}_{\mathrm{o}}$ and $\mathcal{R}_{\mathrm{e}}$ are two body-fixed frames attached, respectively, at the segment base O[1], and at its end-tip E. The segment motion is described in frame $\mathcal{R}_{\mathrm{o}}$ attached to O, which is not necessarily its CoM. The inertial position of O is denoted $\mathbf{r}_{\mathrm{o}}$. Regarding the angular motion, it can be parameterized either by a quaternion or by a set of Euler angles, among others. Alternative parameters have been described in the literature review and are thoroughly covered in (Hughes, 1986). The Euler angles $\mathbf{\Psi}$ are mainly used for their physical meaning while the quaternions $\mathcal{Q}$ are preferred for simulation. Using the Euler angles, the segment attitude is given by $\mathbf{\Psi}_{\mathrm{o}} \in \mathbb{R}^3$:

$$\mathbf{\Psi}_{\mathrm{o}} = \begin{bmatrix} \Psi_x & \Psi_y & \Psi_z \end{bmatrix}^\top \tag{4.1}$$

the related direct cosine matrix w.r.t. the inertial frame is given by:

$$\begin{aligned} {}^{\mathrm{I}}R_{\mathrm{o}} &= R_{\text{3-2-1}}(\mathbf{\Psi}) \\ &= \mathrm{Rot}(\Psi_z, Z)\,\mathrm{Rot}(\Psi_y, Y)\,\mathrm{Rot}(\Psi_x, X) \end{aligned}$$

[1] which is chosen as the reference point here to describe the segment dynamics.



(a) (b)

Figure 4.1: Kinematics of a rigid segment; (a) Frames and vectors describing the global motion w.r.t. the reference point O; (b) Relative position $\delta\mathbf{p}$ of any point P w.r.t. its corresponding projection C on the centroidal axis of the segment

The explicit expression of the rotation matrix for this 3-2-1 Euler convention is obtained as:

$$R_{3\text{-}2\text{-}1}(\mathbf{\Psi}) = \begin{bmatrix} c_y\, c_z & -c_x\, s_z + s_x\, s_y\, c_z & s_x\, s_z + c_x\, s_y\, c_z \\ c_y\, s_z & c_x\, c_z + s_x\, s_y\, s_z & -s_x\, c_z + c_x\, s_y\, s_z \\ -s_y & s_x\, c_y & c_x\, c_y \end{bmatrix} \tag{4.2}$$

where $c_i = \cos(\Psi_i)$, $s_i = \sin(\Psi_i)$ for $i \in \{x; y; z\}$. The alternative expression of the direct cosine matrix using the quaternion is given in Appendix A.2. In the sequel, the free, or unconstrained, coordinates of a free rigid body w.r.t. the inertial frame $\mathcal{R}_\mathrm{I}$ are merged into the states $\mathbf{x}_\mathrm{o}$:

$$\mathbf{x}_\mathrm{o} = \begin{bmatrix} \mathbf{r}_\mathrm{o}^\top & \mathbf{\Psi}_\mathrm{o}^\top \end{bmatrix}^\top \tag{4.3}$$

For any point P along the rigid segment, its relative position w.r.t. O is given by a longitudinal term $\mathbf{p}_c$, which follows the centroidal axis, and by a transverse one $\delta\mathbf{p}$, giving its position in the section plane. Its projection on the centroidal axis, C, is parameterized by the curvilinear abscissa s, while the transverse term is only function of the relative coordinates in the section plane (y, z). This decomposition is illustrated in Figure 4.1. Denoting by "$\mathbf{r}$" and "$\mathbf{p}$" the inertial and relative positions, it yields:

$$\begin{aligned} \mathbf{r}(\mathbf{p}, t) &= \mathbf{r}_\mathrm{o}(t) + \mathbf{p} \\ &= \mathbf{r}_\mathrm{o}(t) + (\mathbf{p}_c(s) + \delta\mathbf{p}(y, z)) \end{aligned} \tag{4.4}$$

4.1.2 Kinetics

When the previous relations are derived w.r.t. time in the inertial frame $\mathcal{R}_\mathrm{I}$, the linear and angular velocities are obtained and merged into the *twist* vector $\mathbf{t}$:

$$\mathbf{t}_\mathrm{o} = \begin{bmatrix} \boldsymbol{v}_\mathrm{o} \\ \boldsymbol{\omega}_\mathrm{o} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & R_{\boldsymbol{\Psi}_\mathrm{o}} \end{bmatrix}}_{\triangleq R_{\mathbf{x}_\mathrm{o}}} \dot{\mathbf{x}}_\mathrm{o} \tag{4.5}$$

where $(\boldsymbol{v}_\mathrm{o}, \boldsymbol{\omega}_\mathrm{o})$ are the linear and angular velocities w.r.t. the inertial frame $\mathcal{R}_\mathrm{I}$, $R_{\boldsymbol{\Psi}}$ is the transformation matrix introduced to convert the time derivative of the Euler angles into an angular rate[2], and $\dot{}$ denotes the time derivative in $\mathcal{R}_\mathrm{I}$. Its expression is detailed in Appendix A.2.

According to the screw theory, the velocity of any point P along the segment is expressed as a function of $\mathbf{t}_\mathrm{o}$. Since the angular rate $\boldsymbol{\omega}_\mathrm{o}$ remains the same along a rigid body, the corresponding twist is given by:

$$\mathbf{t}_\mathrm{p} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega}_\mathrm{o} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_\mathrm{o} + \boldsymbol{\omega}_\mathrm{o}^\times \mathbf{p} \\ \boldsymbol{\omega}_\mathrm{o} \end{bmatrix} \tag{4.6}$$

where the cross product is denoted by the Gibbs notation, with the skew-symmetric matrix:

$$\mathbf{x} \times \mathbf{y} = \mathbf{x}^\times \mathbf{y} \quad \text{with} \quad \mathbf{x}^\times = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \quad \text{for} \quad \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

The last equation (4.6) can be re-written by introducing the rigid version of the *twist-propagation matrix* $\mathbf{A}_\mathrm{p,o}$ defined by S. K. Saha in (Saha et al., 2013), also called *kinematic model* in (Alazard et al., 2008):

$$\mathbf{t}_\mathrm{p} = \underbrace{\begin{bmatrix} \mathbf{I}_3 & -\mathbf{p}^\times \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix}}_{\triangleq \mathbf{A}_\mathrm{p,o}(\mathbf{p})} \mathbf{t}_\mathrm{o} \quad \Longrightarrow \quad \mathbf{t}_\mathrm{e} = \underbrace{\begin{bmatrix} \mathbf{I}_3 & -\mathbf{p}_\mathrm{e}^\times \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix}}_{\triangleq \mathbf{A}_\mathrm{e,o}} \mathbf{t}_\mathrm{o} \tag{4.7}$$

where the end-tip twist, denoted $\mathbf{t}_\mathrm{e}$, is given on the *Right Hand Side (RHS)* by evaluating the twist propagation matrix for the curvilinear abscissa $s = s_\mathrm{e}$. The subscript "e" is used in the sequel to denote the quantities related to the segment end-tip.

The global kinetic energy is now obtained under its classic form by integrating it over the

[2]If a quaternion is used instead of the Euler angles, the free coordinates are given by $\mathbf{x}_\mathrm{o} = \begin{bmatrix} \mathbf{r}_\mathrm{o}^\top & \mathcal{Q}_\mathrm{o}^\top \end{bmatrix}^\top$ with $\mathcal{Q}_\mathrm{o} \in \mathbb{R}^4$, and a similar transformation matrix $R_\mathcal{Q}$ between its time derivative and the angular rate is derived in Appendix A.2.

whole segment. As illustrated in Figure 4.1a, the whole segment $\mathcal{L}$ is assumed to have an elbow-shape with a short rigid portion at the beginning, denoted by $\mathcal{E}$, and a long slender beam for the main part, denoted by $\mathcal{B}$. The rigid kinetic energy T_{rr} is expressed by using the upper row of $\mathbf{A}_{\text{p,o}}$:

$$\begin{aligned} T_{\text{rr}} &= \frac{1}{2}\int_{\mathcal{L}} \boldsymbol{v}^{\top}\boldsymbol{v}\mathrm{d}m \\ &= \frac{1}{2}\,\mathbf{t}_{\text{o}}^{\top}\begin{bmatrix} \int_{\mathcal{L}}\mathbf{I}_3\mathrm{d}m & \int_{\mathcal{L}} -\mathbf{p}^{\times}\mathrm{d}m \\ * & \int_{\mathcal{L}}\left(-\left(\mathbf{p}^{\times}\right)^2\right)\mathrm{d}m \end{bmatrix}\mathbf{t}_{\text{o}} \\ &= \frac{1}{2}\,\mathbf{t}_{\text{o}}^{\top}\underbrace{\begin{bmatrix} m\mathbf{I}_3 & -\mathbf{c}^{\times} \\ * & \mathbf{I}_{\text{o}} \end{bmatrix}}_{\triangleq\mathbf{M}_{\text{r}}}\mathbf{t}_{\text{o}} \end{aligned} \tag{4.8}$$

where m denotes the segment's mass; $\mathbf{c} = \int_{\mathcal{L}}\mathbf{p}\mathrm{d}m$ is the *first moment of inertia* defined by $\mathbf{c}/m = \mathbf{p}_G$, with $\mathbf{p}_G$ the relative position of its CoM w.r.t. O in $\mathcal{R}_{\text{o}}$; and $\mathbf{I}_{\text{o}} = \int_{\mathcal{L}}(-\left(\mathbf{p}^{\times}\right)^2)\mathrm{d}m$ is the *second moment of inertia* of the undeformed body, expressed at O in $\mathcal{R}_{\text{o}}$ (Hughes, 1986; Spong et al., 2006). The symbol "$*$" is introduced to represent the symmetric parts of a matrix.

Besides, if a hub were to be considered, its mass matrix at O expressed in $\mathcal{R}_{\text{o}}$ must be added to $\mathbf{M}_{\text{r}}$. According to the previous expression in (4.8), its mass matrix is given as follows for a rigid hub :

$$\mathbf{M}_{hub} = \begin{bmatrix} m_{hub}\mathbf{I}_3 & -m_{hub}\mathbf{p}_{G_{hub}}^{\times} \\ * & \mathbf{I}_{hub,O} \end{bmatrix}$$

4.1.3 Dynamics

The rigid segment dynamics are derived using the *quasi-Lagrangian equations* introduced in (Hughes, 1986). In this formalism, the twist variables are considered as the velocity states, instead of the time derivatives of the generalized coordinates. As mentioned in the literature review, this step simplifies greatly the rotational dynamics. Indeed, in the classic Lagrangian theory, the generalized coordinates $\mathbf{x}$ and their corresponding velocities $\dot{\mathbf{x}}$ are used to describe the motion. Under matrix form, dynamics is commonly written as (Spong et al., 2006):

$$\boldsymbol{Q}_{\mathbf{x}} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\mathbf{x}}}\right) - \frac{\partial L}{\partial \mathbf{x}} \tag{4.9}$$

where $\boldsymbol{Q}_{\mathbf{x}}$ are the generalized efforts related to $\mathbf{x}$, and L is the Lagrangian. This latter is defined as the difference between the kinetic energy T and the potential energy U:

$$L(\mathbf{x}, \dot{\mathbf{x}}) = T(\mathbf{x}, \dot{\mathbf{x}}) - U(\mathbf{x}) \tag{4.10}$$

The expression of (4.9) becomes quite intricate to derive for the rotational motion, because the corresponding kinetic energy depends simultaneously on the generalized coordinates and on their velocities:

$$T_{\text{rot}}(\boldsymbol{\Psi}, \dot{\boldsymbol{\Psi}}) \;=\; \frac{1}{2}\,\boldsymbol{\omega}^{\top}\,\mathbf{I}\,\boldsymbol{\omega} \;=\; \frac{1}{2}\,\dot{\boldsymbol{\Psi}}^{\top}\,R_{\boldsymbol{\Psi}}^{\top}\,\mathbf{I}\,R_{\boldsymbol{\Psi}}\,\dot{\boldsymbol{\Psi}}$$

To overcome this limitation, the quasi-Lagrangian equations are preferred to describe the rigid-body motion. Introduced in (Hughes, 1986), they use the generalized coordinates and the twist $\mathbf{t}$ to express dynamics. The angular rate of the reference frame $\boldsymbol{\omega}$ serves as generalized rotational speed and provides more compact equations. The Lagrangian is re-written as:

$$L(\mathbf{x}, \mathbf{t}) = T(\mathbf{x}, \mathbf{t}) - U(\mathbf{x})$$

And the state equations are given under matrix form by:

$$\boldsymbol{Q} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \mathbf{t}}\right) - \frac{\tilde{\partial} L}{\partial \mathbf{x}} \tag{4.11}$$

to summarize the following set of equations (Hughes, 1986; Meirovitch, 1991):

$$\boldsymbol{Q}_{v} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \boldsymbol{v}}\right) - \left(\frac{\partial L}{\partial \mathbf{r}}\right) \tag{4.12a}$$

$$\boldsymbol{Q}_{\omega} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \boldsymbol{\omega}}\right) - R_{\boldsymbol{\Psi}}^{-\top}\left(\frac{\partial L}{\partial \boldsymbol{\Psi}}\right) + \boldsymbol{v}^{\times}\left(\frac{\partial L}{\partial \boldsymbol{v}}\right) \tag{4.12b}$$

where $\boldsymbol{v}$ is the inertial speed of the reference point, $\boldsymbol{\omega}$ is the angular rate of the reference frame, and $\boldsymbol{Q}$ are the generalized forces corresponding, respectively, to the linear speed with $\boldsymbol{Q}_{\boldsymbol{v}}$, and to the angular rate with $\boldsymbol{Q}_{\boldsymbol{\omega}}$. The term $\frac{\tilde{\partial} L}{\partial \mathbf{x}}$ is an extended version of the derivative w.r.t. segment states. It is denoted by:

$$\frac{\tilde{\partial} L}{\partial \mathbf{x}} \triangleq \begin{bmatrix} \frac{\partial L}{\partial \mathbf{r}} \\ R_{\boldsymbol{\Psi}}^{-\top}\left(\frac{\partial L}{\partial \boldsymbol{\Psi}}\right) - \boldsymbol{v}^{\times}\left(\frac{\partial L}{\partial \boldsymbol{v}}\right) \end{bmatrix} \tag{4.13}$$

N.B.: The equations provided in (4.12) are slightly different from the ones available in (Hughes, 1986; Meirovitch, 1991). Indeed, the terms $\boldsymbol{\omega} \times \frac{\partial L}{\partial \boldsymbol{v}}$ and $\boldsymbol{\omega} \times \frac{\partial L}{\partial \boldsymbol{\omega}}$ in the linear/angular dynamics have been removed because the time derivative $\frac{\mathrm{d}}{\mathrm{d}t}$ are considered in the inertial frame here, whereas they were expressed in the body-fixed frame in both references. More details are given in (4.14), with the introduction of the notation $\dot{\mathbf{x}}$ and $\mathring{\mathbf{x}}$ to denote, respectively, the time derivatives with respect to the inertial frame $\mathcal{R}_\mathrm{I}$ and w.r.t. the local frame $\mathcal{R}_\mathrm{o}$.

These equations are now derived for the rigid segment considered above. The Lagrangian is only defined by the kinetic energy T, while there is no potential energy U under the weightlessness hypothesis[3]. Before deriving the dynamic equations, the time derivative convention of P. C. Hughes is introduced based on the *vectrix* concept (Hughes, 1986). A differentiation w.r.t. the inertial frame $\mathcal{R}_\mathrm{I}$ is denoted by $\dot{\mathbf{x}}$, while a differentiation w.r.t. the local frame $\mathcal{R}_\mathrm{o}$ is denoted by $\mathring{\mathbf{x}}$. It implies the following relations for a vector $\mathbf{x}$ and a matrix $\mathbf{A}$, considering a local frame rotating along $\boldsymbol{\omega}$:

$$\dot{\mathbf{x}} = \mathring{\mathbf{x}} + \boldsymbol{\omega}^\times \mathbf{x} \tag{4.14a}$$

$$\dot{\mathbf{A}} = \mathring{\mathbf{A}} + \boldsymbol{\omega}^\times \mathbf{A} - \mathbf{A}\,\boldsymbol{\omega}^\times \tag{4.14b}$$

Applying (4.11) with the kinetic energy of a rigid body in (4.8), the state derivatives simplifies because the lagrangian does not depend explicitly upon the reference frame position $\mathbf{r}_\mathrm{o}$ and attitude $\boldsymbol{\Psi}_\mathrm{o}$:

$$\begin{aligned}
\boldsymbol{Q} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{M}_\mathrm{r}\mathbf{t}_\mathrm{o}\right) + \begin{bmatrix} \mathbf{0}_{3\times 1} \\ \boldsymbol{v}_\mathrm{o}^\times \left(m\boldsymbol{v}_\mathrm{o} - \mathbf{c}^\times \boldsymbol{\omega}_\mathrm{o}\right) \end{bmatrix} \\
&= \mathbf{M}_\mathrm{r}\dot{\mathbf{t}}_\mathrm{o} + \dot{\mathbf{M}}_\mathrm{r}\mathbf{t}_\mathrm{o} + \begin{bmatrix} \mathbf{0}_{3\times 1} \\ \left(\mathbf{c}^\times \boldsymbol{\omega}_\mathrm{o}\right)^\times \boldsymbol{v}_\mathrm{o} \end{bmatrix} \\
&= \mathbf{M}_\mathrm{r}\dot{\mathbf{t}}_\mathrm{o} + \left(\mathring{\mathbf{M}}_\mathrm{r} + \boldsymbol{\Omega}_\mathrm{o}\mathbf{M}_\mathrm{r} - \mathbf{M}_\mathrm{r}\boldsymbol{\Omega}_\mathrm{o}\right)\mathbf{t}_\mathrm{o} + \begin{bmatrix} \mathbf{0}_{3\times 1} \\ \left(\mathbf{c}^\times \boldsymbol{\omega}_\mathrm{o}\right)^\times \boldsymbol{v}_\mathrm{o} \end{bmatrix}
\end{aligned}$$

where the cross product property $\mathbf{x}^\times\mathbf{x} = \mathbf{0}$ is used, and where $\boldsymbol{\Omega}_\mathrm{o} = \mathrm{diag}(\boldsymbol{\omega}_\mathrm{o}^\times, \boldsymbol{\omega}_\mathrm{o}^\times)$, with $\mathrm{diag}(X)$ denoting the diagonal or block-diagonal matrix with the elements X (either a set of scalars or matrices).

Since the mass matrix $\mathbf{M}_\mathrm{r}$ does not depend explicitly upon time when expressed in the local

[3]If the gravity field $\mathbf{g}$ were to be considered, instead of introducing it through potential energy, it is preferred to set the base acceleration to $\dot{\mathbf{v}}_o = \mathbf{g}$ in the inverse dynamics of Algorithm 4.1 in order to improve numerical efficiency (see (Walker and Orin, 1982)).

frame $\mathcal{R}_\mathrm{o}$, the term $\mathring{\mathbf{M}}_\mathrm{r}$ vanishes. In addition, the relation of Jacobi[4] is used to develop the nested cross product:

$$(\mathbf{a}\times\mathbf{b})^\times = \mathbf{a}^\times\,\mathbf{b}^\times - \mathbf{b}^\times\mathbf{a}^\times$$

It yields:

$$\boldsymbol{Q} = \mathbf{M}_\mathrm{r}\dot{\mathbf{t}}_\mathrm{o} + \begin{bmatrix} m\boldsymbol{\omega}_\mathrm{o}^\times\boldsymbol{v}_\mathrm{o} - \boldsymbol{\omega}_\mathrm{o}^\times\mathbf{c}^\times\boldsymbol{\omega}_\mathrm{o} \\ \boldsymbol{\omega}_\mathrm{o}^\times\mathbf{c}^\times\boldsymbol{v}_\mathrm{o} + \boldsymbol{\omega}_\mathrm{o}^\times\mathbf{I}_\mathrm{o}\boldsymbol{\omega}_\mathrm{o} \end{bmatrix} - \begin{bmatrix} m\boldsymbol{\omega}_\mathrm{o}^\times\boldsymbol{v}_\mathrm{o} - \mathbf{c}^\times\boldsymbol{\omega}_\mathrm{o}^\times\boldsymbol{\omega}_\mathrm{o} \\ \mathbf{c}^\times\boldsymbol{\omega}_\mathrm{o}^\times\boldsymbol{v}_\mathrm{o} + \mathbf{I}_\mathrm{o}\boldsymbol{\omega}_\mathrm{o}^\times\boldsymbol{\omega}_\mathrm{o} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{c}^\times\boldsymbol{\omega}_\mathrm{o}^\times\boldsymbol{v}_\mathrm{o} - \boldsymbol{\omega}_\mathrm{o}^\times\mathbf{c}^\times\boldsymbol{v}_\mathrm{o} \end{bmatrix}$$

which further simplifies, introducing the matrix $\mathbf{E}_v = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix}$ from (Saha et al., 2013):

$$\begin{aligned} \boldsymbol{Q} &= \mathbf{M}_\mathrm{r}\dot{\mathbf{t}}_\mathrm{o} + \begin{bmatrix} -\boldsymbol{\omega}_\mathrm{o}^\times\mathbf{c}^\times\boldsymbol{\omega}_\mathrm{o} \\ \boldsymbol{\omega}_\mathrm{o}^\times\mathbf{I}_\mathrm{o}\boldsymbol{\omega}_\mathrm{o} \end{bmatrix} \\ &= \mathbf{M}_\mathrm{r}\dot{\mathbf{t}}_\mathrm{o} + \boldsymbol{\Omega}_\mathrm{o}\begin{bmatrix} \mathbf{0}_{3\times3} & -\mathbf{c}^\times \\ \mathbf{0}_{3\times3} & \mathbf{I}_\mathrm{o} \end{bmatrix}\mathbf{t}_\mathrm{o} \\ &= \mathbf{M}_\mathrm{r}\dot{\mathbf{t}}_\mathrm{o} + \boldsymbol{\Omega}_\mathrm{o}\mathbf{M}_\mathrm{r}\mathbf{E}_v\mathbf{t}_\mathrm{o} \end{aligned} \tag{4.15}$$

The generalized efforts are developed in terms of constraint effort applied at the segment's ends. When coming back to the definition of *virtual work* in terms of speed (Parsa, 2007), they read as follows when applied for only one segment:

$$\boldsymbol{Q}_{q_j} = \sum_i \left(\frac{\partial\mathbf{v}_{M_i}}{\partial\dot{q}_j}\right)^\top \boldsymbol{f}_i + \left(\frac{\partial\boldsymbol{\omega}_{M_i}}{\partial\dot{q}_j}\right)^\top \boldsymbol{n}_{M_i} \tag{4.16}$$

where $\dot{q}_j$ denotes the generalized speeds, $(\boldsymbol{f}_i, \boldsymbol{n}_{M_i})$ are the forces and torques acting at (and along) the point M_i, and $(\mathbf{v}_{M_i}, \boldsymbol{\omega}_{M_i})$ are the linear/angular speeds of this point. In the sequel, forces and torques are merged using the *wrench* notation. In the scope of a robotic arm in weightlessness, they are only applied at the end-tips (i.e., at the interface with preceding and following segments), and are denoted:

$$\boldsymbol{w}_\mathrm{o} = \begin{bmatrix} \boldsymbol{f}_\mathrm{o}^\top & \boldsymbol{n}_\mathrm{o}^\top \end{bmatrix}^\top \tag{4.17a}$$

$$\boldsymbol{w}_\mathrm{e} = \begin{bmatrix} \boldsymbol{f}_\mathrm{e}^\top & \boldsymbol{n}_\mathrm{e}^\top \end{bmatrix}^\top \tag{4.17b}$$

where $(\boldsymbol{f}_\mathrm{o}, \boldsymbol{n}_\mathrm{o})$ and $(\boldsymbol{f}_\mathrm{e}, \boldsymbol{n}_\mathrm{e})$ are the efforts applied at the segment base O and end-tip E. If additional external efforts were to be considered, a general derivation of $\boldsymbol{Q}$ is provided in Appendix A.3.1. Denoting by $(\boldsymbol{v}_\mathrm{o}, \boldsymbol{\omega}_\mathrm{o})$ the inertial speed and angular rate at O, the

[4]The Jacobi relation states that: $\mathbf{a}\times(\mathbf{b}\times\mathbf{c}) + \mathbf{b}\times(\mathbf{c}\times\mathbf{a}) + \mathbf{c}\times(\mathbf{a}\times\mathbf{b}) = 0$, so the double cross product reads: $\mathbf{a}\times(\mathbf{b}\times\mathbf{c}) = -\mathbf{b}\times(\mathbf{c}\times\mathbf{a}) - \mathbf{c}\times(\mathbf{a}\times\mathbf{b}) = \mathbf{b}\times(\mathbf{a}\times\mathbf{c}) + (\mathbf{a}\times\mathbf{b})\times\mathbf{c} = (\mathbf{b}^\times\mathbf{a}^\times + (\mathbf{a}\times\mathbf{b})^\times)\,\mathbf{c}$.

corresponding generalized efforts are expressed by:

$$\boldsymbol{Q}_{q_j} = \left(\frac{\partial}{\partial \dot{q}_j} \begin{bmatrix} \boldsymbol{v}_{\mathrm{o}}^{\top} & \boldsymbol{\omega}_{\mathrm{o}}^{\top} \end{bmatrix} \right) \boldsymbol{w}_{\mathrm{o}} + \left(\frac{\partial}{\partial \dot{q}_j} \begin{bmatrix} \boldsymbol{v}_{\mathrm{e}}^{\top} & \boldsymbol{\omega}_{\mathrm{e}}^{\top} \end{bmatrix} \right) \boldsymbol{w}_{\mathrm{e}} \tag{4.18}$$

Replacing the generalized speeds $\dot{q}_j$ by the twist variables $\boldsymbol{v}_{\mathrm{o}}$ and $\boldsymbol{\omega}_{\mathrm{o}}$, and applying the propagation relation in (4.7) for the end-tip, the generalized efforts are obtained as:

$$\begin{aligned} \boldsymbol{Q} &= \left(\frac{\partial \mathbf{t}_{\mathrm{o}}}{\partial \mathbf{t}_{\mathrm{o}}} \right)^{\top} \boldsymbol{w}_{\mathrm{o}} + \left(\frac{\partial \mathbf{t}_{\mathrm{e}}}{\partial \mathbf{t}_{\mathrm{o}}} \right)^{\top} \boldsymbol{w}_{\mathrm{e}} \\ &= \boldsymbol{w}_{\mathrm{o}} + \mathbf{A}_{\mathrm{e,o}}^{\top} \boldsymbol{w}_{\mathrm{e}} \end{aligned} \tag{4.19}$$

For the free body case, this generalized efforts are sometimes denoted $\boldsymbol{W}_{\mathrm{o}}$ in the sequel, and especially in the algorithms, since they represent the wrench of the **resultant net force and torque** expressed at the segment base O.

Rigid dynamics of a single segment

The dynamic equations of a rigid body are finally obtained for any reference point O by combining (4.15) and (4.19):

$$\begin{array}{rcrcrcl} m\,\dot{\boldsymbol{v}}_{\mathrm{o}} & - & \mathbf{c}^{\times}\,\dot{\boldsymbol{\omega}}_{\mathrm{o}} & - & \boldsymbol{\omega}_{\mathrm{o}}^{\times}\,\mathbf{c}^{\times}\,\boldsymbol{\omega}_{\mathrm{o}} & = & \boldsymbol{f}_{\mathrm{o}} + \boldsymbol{f}_{\mathrm{e}} \\ \mathbf{c}^{\times}\,\dot{\boldsymbol{v}}_{\mathrm{o}} & + & \mathbf{I}_{\mathrm{o}}\,\dot{\boldsymbol{\omega}}_{\mathrm{o}} & + & \boldsymbol{\omega}_{\mathrm{o}}^{\times}\,(\mathbf{I}_{\mathrm{o}}\boldsymbol{\omega}_{\mathrm{o}}) & = & \boldsymbol{n}_{\mathrm{o}} + \boldsymbol{n}_{\mathrm{e}} + \mathbf{p}_{\mathrm{e}}^{\times}\,\boldsymbol{f}_{\mathrm{e}} \end{array}$$

or in a more compact form:

$$\mathbf{M}_{\mathrm{r}}\dot{\mathbf{t}}_{\mathrm{o}} + \boldsymbol{\Omega}_{\mathrm{o}}\mathbf{M}_{\mathrm{r}}\mathbf{E}_{v}\mathbf{t}_{\mathrm{o}} = \boldsymbol{w}_{\mathrm{o}} + \mathbf{A}_{\mathrm{e,o}}^{\top}\,\boldsymbol{w}_{\mathrm{e}} \tag{4.20}$$

4.2 Model of a Rigid Multi-body System

The previous dynamic model of a single segment is now used recursively to obtain the dynamics of a whole chain-like manipulator. Kinematics and kinetics are introduced based on a modified version of the DH parameters (Denavit and Hartenberg, 1955). Manipulator dynamics is derived in the general case of n links, each of them being described by the previous model. A Newton-Euler scheme is developed to compute efficiently the inverse dynamics. A closed-form equation is also introduced based on the DeNOC approach, with the derivation of two algorithms to get the global mass matrix and the Coriolis and centrifugal vector. Finally, a simulation scheme is built by solving the forward dynamics using (Saha, 1999; Saha et al., 2013).

Though well known in the literature of rigid systems, these steps are developed for sake of completeness. They allow to introduce the global DeNOC notations and algorithms, before extending them to the flexible case in the Chapter 5. The superscript "r" is used to denote the rigid terms, and "f" will be used later for the flexible ones.

4.2.1 Kinematics

The *Denavit Hartenberg (DH)* parameters are used to describe the kinematic structure of a manipulator. A frame $\mathcal{R}_i$ is rigidly fixed to each segment, and a set of four parameters $\left[\begin{array}{cccc} \theta_i & d_i & a_i & \alpha_i \end{array}\right]$ is used to characterize the homogeneous transformation ${}^{i-1}T_i$ between two successive segments (Craig, 1989; Spong et al., 2006). With the classic parameters, the i^{th} joint axis[5] is aligned with the Z-axis of the previous frame $\mathcal{R}_{i-1}$. In addition, its origin O_i is often located at the segment's end-tip. As mentioned in the literature review, the *modified parameters* were proposed to manage the ambiguous configurations arising for tree-type systems (Khalil and Kleinfinger, 1986). Nevertheless, with these modified parameters, the frame is often located at the elbow, at point A in Figure 4.1a, and not at the segment base, as desired with $\mathcal{R}_{\mathrm{o}}$. Therefore, an adapted version of the DH formalism is proposed in Appendix A.1 to ensure that the frames are at the beginning of the segments for the particular case of elbow shapes[6]. The three formalisms are compared thoroughly on a planar and a 3D examples in Appendix A.1. The resulting frames are illustrated for the Canadarm example in Figure 4.2.

Using this adapted formalism, kinematics are still described by a set of four DH parameters, but taken in the following order $\left[\begin{array}{cccc} d_i & a_i & \alpha_i & \theta_i \end{array}\right]$. In the rigid case, the corresponding homogeneous transformation ${}^{i-1}T_i^{\mathrm{r}}$ between the successive frames $\mathcal{R}_{i-1} \triangleq \mathcal{R}_{\mathrm{o}_{i-1}}$ and $\mathcal{R}_i \triangleq \mathcal{R}_{\mathrm{o}_i}$ is performed by:

$$
\begin{aligned}
{}^{i-1}T_i^{\mathrm{r}} &= \mathrm{Trans}(d_i, Z)\,\mathrm{Trans}(a_i, X)\,\mathrm{Rot}(\alpha_i, X)\,\mathrm{Rot}(\theta_i, Z) \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

[5] i.e., the *rotation axis* for a revolute joint, or the *translation axis* for a prismatic one.

[6] It is worth noticing that this formalism is not well suited for prismatic joints because the i^{th} variable position d_i would be accounted for in the $(i+1)^{th}$ set of parameters. If one considers prismatic joints, the modified parameters remain the best choice.

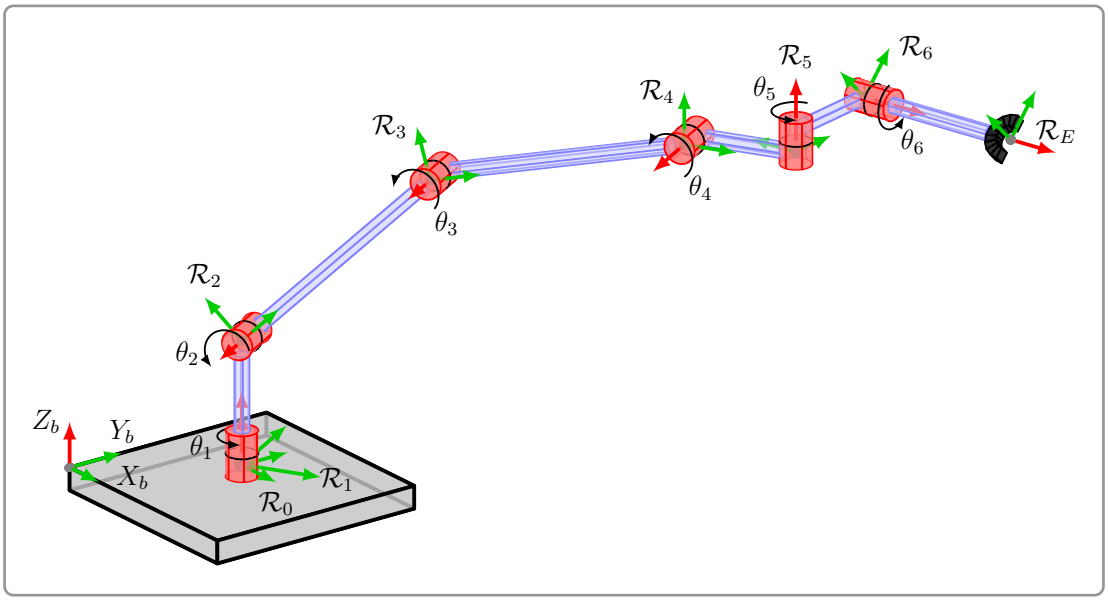


Figure 4.2: Adapted Denavit-Hartenberg frames for the Canadarm example

leading to:

$$ {}^{i-1}T_i^{\mathrm{r}} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & a_i \\ c_{\alpha_i} s_{\theta_i} & c_{\alpha_i} c_{\theta_i} & -s_{\alpha_i} & 0 \\ s_{\alpha_i} s_{\theta_i} & s_{\alpha_i} c_{\theta_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.21} $$

where c_x and s_x denote respectively $\cos(x)$ and $\sin(x)$. The rotation matrix between the two frames is given by ${}^{i-1}T_i^{\mathrm{r}}(1{:}3, 1{:}3)$[7]:

$$ \begin{aligned} {}^{i-1}R_i^{\mathrm{r}}(\theta_i) &= \mathrm{Rot}(\alpha_i, X)\,\mathrm{Rot}(\theta_i, Z) \\ &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 \\ c_{\alpha_i} s_{\theta_i} & c_{\alpha_i} c_{\theta_i} & -s_{\alpha_i} \\ s_{\alpha_i} s_{\theta_i} & s_{\alpha_i} c_{\theta_i} & c_{\alpha_i} \end{bmatrix} \end{aligned} \tag{4.22} $$

and the vector relying them, between O_{i-1} and $E_{i-1} \equiv O_i$, is given by ${}^{i-1}T_i^{\mathrm{r}}(1{:}3, 4)$:

$$ \begin{aligned} \mathbf{p}_{i-1}^{\mathrm{r}} \triangleq \mathbf{p}_{\mathrm{e}_{i-1}} &= \mathrm{Trans}(d_i, Z)\,\mathrm{Trans}(a_i, X) \\ &= \begin{cases} {}^{(i-1)}\begin{bmatrix} a_i \\ 0 \\ d_i \end{bmatrix} \\ \quad \text{or} \\ {}^{(i)}\begin{bmatrix} a_i\, c_{\theta_i} + d_i\, s_{\theta_i}\, s_{\alpha_i} \\ -a_i\, s_{\theta_i} + d_i\, c_{\theta_i}\, s_{\alpha_i} \\ d_i\, c_{\alpha_i} \end{bmatrix} \end{cases} \end{aligned} \tag{4.23} $$

[7]Using the Matlab vector notation, with $X(i{:}j)$ being the the sub-vector of X from the i^{th} to the j^{th} component.

where the vector $\mathbf{x}$ expressed in frame $\mathcal{R}_i$ is denoted by ${}^{(i)}\mathbf{x}$.

Considering that each segment has one rigid degree of motion along its joint axis, its generalized coordinate reduces to a scalar term $q_i \in \mathbb{R}$. For a manipulator with n segments, the global vector of generalized coordinates is given by:

$$\mathbf{q}_\mathrm{r} = \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix}^\top \tag{4.24}$$

with $q_i = \begin{bmatrix} \theta_i \end{bmatrix}$ (resp. $\begin{bmatrix} d_i \end{bmatrix}$) for a revolute (resp. prismatic) joint, while the three remaining DH parameters are constant.

Using recursively the previous transformations, the inertial position of any point M along the i^{th} link, with the relative coordinates ${}^i\mathbf{p}_M$ in $\mathcal{R}_i$, is then given in the base frame $\mathcal{R}_\mathrm{o}$ as a function of $\mathbf{q}_\mathrm{r}$. Provided that $\tilde{\mathbf{x}}$ denotes the homogeneous form of a position vector (i.e., $\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x}^\top & 1 \end{bmatrix}^\top$), it yields:

$$\tilde{\mathbf{r}}_M = \left({}^0T^\mathrm{r}_1(q_1)\, {}^1T^\mathrm{r}_2(q_2) \dots {}^{i-1}T^\mathrm{r}_i(q_i) \right) {}^i\tilde{\mathbf{p}}_M = {}^0T^\mathrm{r}_i(\mathbf{q}_\mathrm{r})\, {}^i\tilde{\mathbf{p}}_M \tag{4.25}$$

And especially for the end-effector, both its position and orientation can be described by an additional frame, commonly known as the *effector frame* $\mathcal{R}_\mathrm{E} \equiv \mathcal{R}_{n+1}$. The additional transformation ${}^nT_\mathrm{E}$ gives the kinematic properties of a given tool fixed at the end-effector $\mathrm{E} = O_{n+1}$. More specifically, its relative position is described by $\mathbf{p}^\mathrm{r}_{n+1} \triangleq {}^n\mathbf{p}_\mathrm{e}$, and its orientation by the rotation matrix ${}^nR^\mathrm{r}_{n+1} \triangleq {}^nR_\mathrm{E}$. Denoting the global transformation by ${}^0T^\mathrm{r}_E = {}^0T^\mathrm{r}_n(\mathbf{q}_\mathrm{r})\, {}^nT^\mathrm{r}_\mathrm{E}$, which depends upon $\mathbf{q}_\mathrm{r}$, the effector frame is obtained by:

$$\mathbf{r}_\mathrm{E}(\mathbf{q}_\mathrm{r}) = {}^0T^\mathrm{r}_E \begin{bmatrix} \mathbf{0}_{3\times1} \\ 1 \end{bmatrix} \qquad \text{and} \qquad {}^0R^\mathrm{r}_\mathrm{E}(\mathbf{q}_\mathrm{r}) = {}^0T^\mathrm{r}_\mathrm{E}(1{:}3, 1{:}3) \tag{4.26}$$

As a conclusion, a robotic arm is described by two dual quantities: the position and the orientation of its end-effector, $\mathbf{x}_\mathrm{E}$, or the angles between its segments, $\mathbf{q}_\mathrm{r}$. These latter are called the *joint angles* or the *arm configuration*. Kinematics is based on the last relation linking the end-effector position and orientation to the joint angles. It can be summarized as:

$$\mathbf{x}_\mathrm{E} = \begin{bmatrix} \mathbf{r}_\mathrm{E} \\ \boldsymbol{\Psi}_\mathrm{E} \end{bmatrix} = f_{kin}(\mathbf{q}_\mathrm{r}) \tag{4.27}$$

4.2.2 Kinetics

When differentiating the previous kinematic relations with respect to time, one derives the robot kinetics, required to express its kinetic energy and then to derive its dynamics. A recursive relation between the twists of two successive segment is derived first. This relation is necessary to apply the segment dynamics in (4.20) which depends upon the reference twist $\mathbf{t}_{\mathrm{o}}$. The recursive twist relation is straightforward by adding the angular (resp. linear) speed coming from the revolute (resp. prismatic) joint motion to the end-tip twist of the previous segment:

$$\begin{aligned}\mathbf{t}_{\mathrm{o}_i} &= \mathbf{t}_{\mathrm{e}_{i-1}} + \dot{q}_i\,\mathbf{z}_i \\ &= \mathbf{A}_{\mathrm{e}_{i-1},\mathrm{o}_{i-1}}\,\mathbf{t}_{\mathrm{o}_{i-1}} + \dot{q}_i\,\mathbf{z}_i\end{aligned}$$

where $\mathbf{z}_i$ is the generalized joint axis[8], and $\mathbf{A}_{\mathrm{e}_{i-1},\mathrm{o}_{i-1}}$ is the twist-propagation matrix in (4.7) written for the end-tip of the $(i-1)^{th}$ segment.

To alleviate the notations in the sequel, the i^{th} reference twist is denoted $\mathbf{t}_i \triangleq \mathbf{t}_{\mathrm{o}_i}$, and the twist-propagation matrix from the $(i-1)^{th}$ to the i^{th} segment is denoted $\mathbf{A}_{i,i-1}$. With these notations, the recursive twist relation reads:

$$\boxed{\mathbf{t}_i = \mathbf{A}_{i,i-1}\,\mathbf{t}_{i-1} + \mathbf{P}_i\,\dot{q}_i} \tag{4.28}$$

where $\mathbf{A}_{i,i-1}$ and $\mathbf{P}_i$ are the rigid version of the *twist propagation matrix*, and of the *joint rate propagation matrix*. Reminding the expression of $\mathbf{p}^{\mathrm{r}}_{i-1} \triangleq \mathbf{p}_{\mathrm{e}_{i-1}}$ in (4.23), they read as follows (Mohan and Saha, 2007):

$$\mathbf{A}^{\mathrm{r}}_{i,i-1} = \begin{bmatrix} \mathbf{I}_3 & -\left(\mathbf{p}^{\mathrm{r}}_{i-1}\right)^{\times} \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 \end{bmatrix} \tag{4.29a}$$

$$\mathbf{P}^{\mathrm{r}}_i = \begin{bmatrix} \mathbf{z}_i \end{bmatrix} \tag{4.29b}$$

These two matrices are introduced to obtained the kinetic relations in a compact matrix form, and they will be easily extended to the flexible case since the same set of equations holds. In (4.28), it is very important to notice that the twist $\mathbf{t}_i$ and $\mathbf{t}_{i-1}$ may be expressed in, respectively, $\mathcal{R}_i$ and $\mathcal{R}_{i-1}$ to speed up the computations by limiting the changes of frame. To that end, $\mathbf{P}_i$ is rather expressed in $\mathcal{R}_i$, where its expression is simpler, while $\mathbf{A}_{i,i-1}$ is augmented by the change of frame, because it is expressed in $\mathcal{R}_{i-1}$ to perform the product with

[8] $\mathbf{z}_i = \begin{bmatrix} \mathbf{0}^\top & Z_i^\top \end{bmatrix}^\top$ for a revolute joint, and $\mathbf{z}_i = \begin{bmatrix} Z_i^\top & \mathbf{0}^\top \end{bmatrix}^\top$ for a prismatic one. With the adapted DH convention, Z_i reads simply $[0, -s_{\alpha_i}, c_{\alpha_i}]^\top$ in $\mathcal{R}_{i-1}$, or $[0,0,1]^\top$ in $\mathcal{R}_i$.

$\mathbf{t}_{i-1}$. Therefore, the generalized rotation matrix ${}^{i-1}\boldsymbol{R}_i$ is introduced and the final expressions of $\mathbf{A}_{i,i-1}$ and $\mathbf{P}_i$ are given by:

$$\mathbf{A}_{i,i-1} = {}^{i-1}\boldsymbol{R}_i^{\top}\ {}^{(i-1)}\mathbf{A}^{\mathrm{r}}_{i,i-1} \tag{4.30a}$$

$$\mathbf{P}_i = {}^{(i)}\mathbf{P}^{\mathrm{r}}_i \tag{4.30b}$$

where the generalized rotation matrix is given by:

$${}^{i-1}\boldsymbol{R}_i = \begin{bmatrix} {}^{i-1}R^{\mathrm{r}}_i & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & {}^{i-1}R^{\mathrm{r}}_i \end{bmatrix} \tag{4.31}$$

The relation (4.28) is the recursive version of the *Decoupled Natural Orthogonal Complement (DeNOC)* approach, and will be used to perform efficient simulations. Nonetheless, kinetics may also be viewed globally by seeing the manipulator as a whole, in order to derive the arm Jacobian.

Jacobian Matrix by the DENOC Approach To express the global twist of the segments $\mathbf{t} = [\,\mathbf{t}_1^{\top}, \ldots \mathbf{t}_n^{\top}\,]^{\top}$ as a function of the global generalized coordinate $\mathbf{q}_{\mathrm{r}} = [\,q_1, \ldots q_n\,]^{\top}$, J. Angeles and S. K. Lee introduced the NOC matrix $\mathbf{N}$ such that $\mathbf{t} = \mathbf{N}\,\dot{\mathbf{q}}_{\mathrm{r}}$. This denomination has its roots into the holonomic constraints expression. Describing the fact that the segments of a manipulator are forced to only move along one axis, these kinematic constraints can be written as a function of the segment coordinates $\mathbf{x} = [\,\mathbf{x}_1^{\top}, \ldots, \mathbf{x}_n^{\top}\,]^{\top}$, with $\mathbf{x}_i = \mathbf{x}_{\mathrm{o}_i} = [\,\mathbf{r}_{\mathrm{o}_i}^{\top}, \boldsymbol{\Psi}_{\mathrm{o}_i}^{\top}\,]^{\top}$ (Dimitrov, 2005):

$$\Phi(\mathbf{x}) = 0$$

When differentiated with respect to time, the resulting relation shows that the segment velocities must lie within the null space of the Jacobian matrix of the constraints, which is denoted $\Phi_{\mathbf{x}} = \partial\Phi/\partial\mathbf{x}$. The differentiation yields: $\Phi_{\mathbf{x}}\,\dot{\mathbf{x}} = \mathbf{0}$.

In addition, it has been shown in (4.5) that $\dot{\mathbf{x}}_i$ is linked to the twist $\mathbf{t}_i$ by an invertible matrix $R_{\mathbf{x}_i}$. When this relation is extended to express the global velocity as a function of the global twist, it yields: $\mathbf{t} = [R_{\mathbf{x}}]\,\dot{\mathbf{x}}$, with $[R_{\mathbf{x}}] = \mathrm{diag}\,(R_{\mathbf{x}_i})$ still invertible in the absence of singular configurations. The time differentiation of the constraints can then be re-written as (Dimitrov, 2005):

$$\Phi_{\mathbf{x}}\,\dot{\mathbf{x}} = \mathbf{0} \quad \Longrightarrow \quad \Phi_{\mathbf{x}}\left[R_{\mathbf{x}}^{-1}\right]\mathbf{t} = \mathbf{0}$$

Denoting $\Phi_{\mathbf{t}} \triangleq \Phi_{\mathbf{x}} \left[R_{\mathbf{x}}^{-1}\right]$ and replacing the global twist $\mathbf{t}$ by the NOC relation, it yields:

$$\Phi_{\mathbf{t}}\mathbf{N}\dot{\mathbf{q}}_{\mathrm{r}} = \mathbf{0} \quad \Longrightarrow \quad \Phi_{\mathbf{t}}\mathbf{N} = \mathbf{0}$$

since the generalized coordinates $\mathbf{q}_{\mathrm{r}}$ are, by definition, independent. This last relation states that the columns of $\mathbf{N}$ span the null space of $\Phi_{\mathbf{t}}$, describing the *Space of Allowable Motion* introduced in (Dimitrov, 2005). Each column of $\mathbf{N}$ is therefore orthogonal to each row of $\Phi_{\mathbf{t}}$, justifying its name of NOC.

S. K. Saha introduced the *decoupled* form of $\mathbf{N}$ by using the recursive twist relation in (4.28). As shown in (Saha, 1997), the explicit expression of $\mathbf{N}$ can be obtained by expressing first the twist relation under matrix form, in order to highlight the kinematic structure of the manipulator. Starting with the base twist $\mathbf{t}_0 = \mathbf{0}$ for a fixed base, the following system of equations is obtained for n segments:

$$\begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{t}_3 \\ \vdots \\ \mathbf{t}_n \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{2,1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{3,2} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{n,n-1} & \mathbf{0} \end{bmatrix}}_{\triangleq \mathbf{A}} \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{t}_3 \\ \vdots \\ \mathbf{t}_n \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{P}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_3 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{P}_n \end{bmatrix}}_{\triangleq \mathbf{N}_d} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

such that

$$\begin{bmatrix} \mathbf{I}_6 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{A}_{2,1} & \mathbf{I}_6 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}_{3,2} & \mathbf{I}_6 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{A}_{n,n-1} & \mathbf{I}_6 \end{bmatrix} \mathbf{t} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_3 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_n \end{bmatrix} \dot{\mathbf{q}}_{\mathrm{r}}$$

In the *Left Hand Side (LHS)*, the matrix $(\mathbf{I} - \mathbf{A})$ can be inverted analytically using the Gaussian elimination technique. Denoting $\mathbf{N}_l$ its inverse and using the transitivity notation: $\mathbf{A}_{i,j} = \mathbf{A}_{i,i-1}\,\mathbf{A}_{i-1,j}$, the system of equations reduces to:

$$\mathbf{t} = \underbrace{\begin{bmatrix} \mathbf{I}_6 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{2,1} & \mathbf{I}_6 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{3,1} & \mathbf{A}_{3,2} & \mathbf{I}_6 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \cdots & \mathbf{0} \\ \mathbf{A}_{n,1} & \mathbf{A}_{n,2} & \cdots & \mathbf{A}_{n,n-1} & \mathbf{I}_6 \end{bmatrix}}_{(\mathbf{I}-\mathbf{A})^{-1} \triangleq \mathbf{N}_l} \underbrace{\begin{bmatrix} \mathbf{P}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_3 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_n \end{bmatrix}}_{\triangleq \mathbf{N}_d} \dot{\mathbf{q}}_\mathrm{r} \tag{4.32}$$

It is worth noticing that this relation is actually the Jacobian matrix[9] written for every segment twist. This matrix is characterized by the linear dependency with the joint speeds:

$$\mathbf{t}_i = \left[\begin{array}{ccc|c|ccc} \mathbf{A}_{i,1} & \cdots & \mathbf{A}_{i,i-1} & \mathbf{I}_6 & \mathbf{0} & \cdots & \mathbf{0} \end{array} \right] \mathbf{N}_d \, \dot{\mathbf{q}}_\mathrm{r}$$

For example, the effector Jacobian matrix $\mathbf{J}_\mathrm{E}$ is obtained by introducing the additional twist-propagation matrix $\mathbf{A}_{n+1,n}$, corresponding to the end-effector position $\mathbf{p}^\mathrm{r}_{n+1} \triangleq {}^n\mathbf{p}_\mathrm{e}$ expressed w.r.t. the last segment frame $\mathcal{R}_n$:

$$\mathbf{t}_\mathrm{E} = \underbrace{\left[\begin{array}{ccccc} \mathbf{A}_{n+1,1} & \mathbf{A}_{n+1,2} & \cdots & \mathbf{A}_{n+1,n-1} & \mathbf{A}_{n+1,n} \end{array} \right] \mathbf{N}_d}_{\triangleq \mathbf{J}_\mathrm{E}} \, \dot{\mathbf{q}}_\mathrm{r} \tag{4.33}$$

It can be summarized by the following kinetic relation, to compare with the kinematic equivalent in (4.27):

$$\mathbf{t}_\mathrm{E} = \begin{bmatrix} \boldsymbol{v}_\mathrm{E} \\ \boldsymbol{\omega}_\mathrm{E} \end{bmatrix} = \mathbf{J}_\mathrm{E}(\mathbf{q}_\mathrm{r}) \, \dot{\mathbf{q}}_\mathrm{r} \tag{4.34}$$

Kinematic Singularities and Manipulability The kinematic singularities $\mathbf{q}_{sing}$ are defined as the configurations where the Jacobian matrix $\mathbf{J}_\mathrm{E}$ is loosing its full rank. In other words, it exists a non-zero vector of joint speeds $\dot{\boldsymbol{\eta}}$ such that: $\mathbf{J}_\mathrm{E}(\mathbf{q}_{sing})\dot{\boldsymbol{\eta}} = \mathbf{0}$. It means that a differential motion in the direction $\dot{\boldsymbol{\eta}}$ does not produce any linear nor angular motion of the end-effector. This situation occurs for example when the axes of two joints are aligned, since they can move in the opposite way without moving the effector. Another classic example is when the manipulator reaches the limits of its workspace: the effector cannot go further

[9] "Jacobian" being understood here in its robotic meaning, i.e., as the kinetic relation between the joint speeds and the twist of any point: $\mathbf{t}_\mathrm{p} = \mathbf{J}_\mathrm{p} \, \dot{\mathbf{q}}$.

and it looses some degrees of motion. This is why this matrix is used to characterize the *manipulability* of a given configuration. It gives the capacity of the effector to move around this configuration before reaching a singularity. A classic index of dexterity is given by the volume of the *manipulability ellipsoid*, i.e., the product of the Jacobian singular values (Spong et al., 2006). In the special case of a non-redundant arm, it reads:

$$\mu(\mathbf{q}_\mathrm{r}) = \sqrt{\det\left(\mathbf{J}_\mathrm{E}(\mathbf{q}_\mathrm{r})\,\mathbf{J}_\mathrm{E}^\top(\mathbf{q}_\mathrm{r})\right)} \tag{4.35}$$

This index highly depends on the units of distance or angle used to describe the robot kinematics, which lead to great variations on the Jacobian matrix. For sake of numerical implementation, different solutions have been proposed in the literature to overcome this limitation: the condition number, for example, allows to use a dimensionless quantity between 0 and 1 based on the singular value decomposition of the Jacobian matrix, while additional transformations are also introduced to make these indices invariant with the choice of the distance unit (Staffetti et al., 2002). These measures are crucial when designing the path planner of the arm, in order to avoid the singularities and to prevent the effector from reaching the workspace limits.

The notation and preliminary concepts of the DeNOC approach have been presented in this kinetics section. It allows to derive the dynamics by two approaches: a recursive one for efficient simulation schemes, and a global one for control purpose. The relation (4.28) is used to derive a recursive Newton-Euler scheme based on the constrained equations of motion of a single segment, while the relation (4.32) is more suited to derive explicitly the closed-form dynamic equation for the whole manipulator.

4.2.3 Constrained Dynamics of the Segments

In order to use the previous dynamic model of a single rigid link in (4.20), the time derivative of the twist is computed recursively using the relation (4.28) :

$$\boxed{\dot{\mathbf{t}}_i = \dot{\mathbf{A}}_{i,i-1}\,\mathbf{t}_{i-1} + \mathbf{A}_{i,i-1}\,\dot{\mathbf{t}}_{i-1} + \dot{\mathbf{P}}_i\,\dot{q}_i + \mathbf{P}_i\,\ddot{q}_i} \tag{4.36}$$

The differentiations of $\mathbf{A}^\mathrm{r}_{i,i-1}$ and $\mathbf{P}^\mathrm{r}_i$ in their respective frame are given by the time derivatives of matrices and vectors in (4.14). Noticing that they are both constant w.r.t., respectively,

$\mathcal{R}_{i-1}$ and $\mathcal{R}_i$, it yields:

$$\begin{aligned}\dot{\mathbf{A}}^{\mathrm{r}}_{i,i-1} &= \boldsymbol{\Omega}_{i-1}\,\mathbf{A}^{\mathrm{r}}_{i,i-1} - \mathbf{A}^{\mathrm{r}}_{i,i-1}\boldsymbol{\Omega}_{i-1} \\ &= \begin{bmatrix} \mathbf{0}_{3\times3} & -\left(\boldsymbol{\omega}_{i-1}\times\mathbf{p}^{\mathrm{r}}_{i-1}\right)^{\times} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \qquad (4.37a)\\ \dot{\mathbf{P}}^{\mathrm{r}}_{i} &= \boldsymbol{\Omega}_i\,\mathbf{P}^{\mathrm{r}}_{i} \qquad (4.37b)\end{aligned}$$

where $\boldsymbol{\Omega}_i = \mathrm{diag}(\boldsymbol{\omega}_i^{\times}, \boldsymbol{\omega}_i^{\times})$. $\boldsymbol{\omega}_i \triangleq \boldsymbol{\omega}_{\mathrm{o}_i}$ denotes the angular rate of the i^{th} segment, and is expressed in $\mathcal{R}_i$. Finally, the change of frame ${}^{i-1}\boldsymbol{R}_i$ in (4.30a) must be applied on the time derivative of $\mathbf{A}^{\mathrm{r}}_{i,i-1}$ since its expression has been derived in $\mathcal{R}_{i-1}$. By doing this, the summation of (4.36) can be performed and expresses $\dot{\mathbf{t}}_i$ in $\mathcal{R}_i$. The time derivatives in (4.36) are finally given by:

$$\dot{\mathbf{A}}_{i,i-1} = {}^{i-1}\boldsymbol{R}_i^{\top}\,{}^{(i-1)}\dot{\mathbf{A}}^{\mathrm{r}}_{i,i-1} \qquad (4.38a)$$

$$\dot{\mathbf{P}}_{i} = \boldsymbol{\Omega}_i\,{}^{(i)}\mathbf{P}^{\mathrm{r}}_{i} \qquad (4.38b)$$

In addition to the twist and its time derivative, the constrained dynamics of the segment is also driven by the efforts applied at both ends. Using the Newton's third law on the i^{th} segment, the wrench at the end-tip is the opposite of the one applied on the next segment :

$$\boldsymbol{w}_{\mathrm{e}_i} = -\boldsymbol{w}_{\mathrm{o}_{i+1}}$$

Denoting by $\boldsymbol{w}_i = \boldsymbol{w}_{\mathrm{o}_i}$ the efforts applied by the $(i-1)^{th}$ segment on the i^{th} one, the equation governing the constrained dynamics of this latter reads :

$$\boxed{\mathbf{M}_{\mathrm{r}_i}\dot{\mathbf{t}}_i + \boldsymbol{\gamma}_{\mathrm{r}_i} = \boldsymbol{w}_i - \mathbf{A}^{\top}_{i+1,i}\,\boldsymbol{w}_{i+1}} \qquad (4.39)$$

where $\boldsymbol{\gamma}_{\mathrm{r}_i}$ denotes the vector of Coriolis and centrifugal terms:

$$\boldsymbol{\gamma}_{\mathrm{r}_i} = \boldsymbol{\Omega}_i\mathbf{M}_{\mathrm{r}_i}\mathbf{E}_v\mathbf{t}_i \qquad (4.40)$$

Any additional external wrench $\boldsymbol{w}_{ext,i}$ can be considered in this equation by merely adding its value expressed at O_i on the RHS of (4.39), as explained in Appendix A.3.1.

N.B.: In the segment dynamics in (4.20) and (4.39), the twist derivative is given w.r.t. the inertial frame $\mathcal{R}_\mathrm{I}$ instead of the floating frame $\mathcal{R}_i$, because it avoids any ambiguities in the sequel. Indeed, the terms in (4.28) are expressed in different frames, so if one expresses the time derivative in the floating-frame, it yields:

$$\mathring{\mathbf{t}}_i \;=\; \left.\frac{d}{dt}\right|_{\mathcal{R}_i} (\mathbf{A}_{i,i-1}\mathbf{t}_{i-1}) + \left.\frac{d}{dt}\right|_{\mathcal{R}_i} (\mathbf{P}_i \dot{q}_i)$$

The time derivative of the second term $\mathbf{P}_i$ w.r.t. $\mathcal{R}_i$ is straightforward since it has a simple expression in this frame, but differentiation of the first term is more demanding since they are easier to express in $\mathcal{R}_{i-1}$. It would imply to introduce the relative angular rate ${}^{i-1}\boldsymbol{\omega}_i$ between both frames. But the angular rates $\boldsymbol{\omega}_i$ are already computed w.r.t. $\mathcal{R}_\mathrm{I}$ through the twists. As a conclusion, these time derivatives have rather been considered w.r.t. this common inertial frame to simplify the computations.

To fully express the inverse dynamics, the final step consists in computing the generalized efforts corresponding to the generalized coordinates $\mathbf{q}_\mathrm{r}$ given in (4.24).

$$\boldsymbol{\tau}_\mathrm{r} = \begin{bmatrix} \tau_1 & \dots & \tau_n \end{bmatrix}^\top \tag{4.41}$$

with τ_i the torque (resp. force) applied by the joint motor of the revolute (resp. prismatic) joint. In the same fashion that the twist is only increased along the joint axis $\mathbf{z}_i$, the motor wrench applied on the segment is only effective along this same axis. By contrast, the constraint wrench is applied in the orthogonal directions. Therefore, the base wrench can be split in two terms:

$$\boldsymbol{w}_i = \boldsymbol{w}_{int,i} + \underbrace{\boldsymbol{w}_{mot,i}}_{\triangleq \tau_i \, \mathbf{z}_i}$$

with $\boldsymbol{w}_{int,i}$ and $\boldsymbol{w}_{mot,i}$ the constraint and motor wrenches applied at the segment's base. Illustrating the fact that internal efforts do not work (Saha et al., 2013), this decomposition yields the following result:

$$\boldsymbol{w}_{int,i} \perp \boldsymbol{w}_{mot,i} \quad\Longrightarrow\quad \boldsymbol{w}_{int,i}^\top \, \mathbf{z}_i = 0$$

and leads to the expression of the i^{th} generalized effort as a function of the i^{th} base wrench[10]:

$$\tau_i = \mathbf{P}_i^\top \, \boldsymbol{w}_i \tag{4.42}$$

[10] An alternative proof of this formula from the viewpoint of velocity constraint is provided in Appendix A.3.1.

In the same fashion, if external efforts are considered at the segment bases, like friction, a set of wrenches $\{\boldsymbol{w}_{ext,k};\, k = 1 \dots n\}$ expressed at O_k is introduced and lead to an additional generalized effort at the i^{th} joint:

$$\boldsymbol{\tau}_{ext,i} = \mathbf{P}_i^\top \left(\sum_{k=i}^{n} \mathbf{A}_{k,i}^\top \boldsymbol{w}_{ext,k} \right) \tag{4.43}$$

Especially for an external effort at the end-effector, this relation yields:

$$\boldsymbol{\tau}_{\mathrm{E},i} = \mathbf{P}_i^\top \, \mathbf{A}_{n+1,i}^\top \, \boldsymbol{w}_{\mathrm{E}} \tag{4.44}$$

These last relations are proven in details in Appendix A.3.1. The latter is also matching the transposed of the i^{th} column of the Jacobian matrix in (4.33), which is consistent with the classic term $\mathbf{J}_{\mathrm{E}}^\top \boldsymbol{w}_{\mathrm{E}}$ arising in the closed-form equation of dynamics derived later in (4.54).

The last equations allow to recursively compute the twist and its time derivative knowing the previous segment ones with (4.28) and (4.36). They also allow to compute the wrench and its related generalized efforts knowing those of the next segment with (4.39) and (4.42). This two recursions together define the Newton-Euler inverse dynamics algorithm presented in the next section.

4.2.4 Inverse Dynamics

The inverse dynamics consists in computing the generalized efforts $\boldsymbol{\tau}_{\mathrm{r}}$ resulting from a given arm configuration $(\mathbf{q}_{\mathrm{r}}, \dot{\mathbf{q}}_{\mathrm{r}}, \ddot{\mathbf{q}}_{\mathrm{r}})$. This function is useful in a twofold sense: it is the basis for efficient algorithms to compute the arm mass matrix and the vector of Coriolis and centrifugal efforts, and it also allows to obtain the efforts to apply at the joints to perform a desired joint trajectory $(\mathbf{q}_{des}, \dot{\mathbf{q}}_{des}, \ddot{\mathbf{q}}_{des})$ in a feedforward control.

In the following, the inverse dynamics is given first by the recursive Newton-Euler algorithm, and then by a global closed-form equation, the terms of which are computed by dedicated DeNOC algorithms. For a rigid robotic arm, this equation takes the classic form of (Spong et al., 2006; Craig, 1989):

$$\mathbf{D}_{\mathrm{r}}(\mathbf{q}_{\mathrm{r}})\, \ddot{\mathbf{q}}_{\mathrm{r}} + \mathbf{h}_{\mathrm{r}}(\mathbf{q}_{\mathrm{r}}, \dot{\mathbf{q}}_{\mathrm{r}}) = \boldsymbol{\tau}_{\mathrm{r}} \tag{4.45}$$

with $\mathbf{D}_{\mathrm{r}}$ the global mass matrix, and $\mathbf{h}_{\mathrm{r}}$ the vector of Coriolis and centrifugal forces.

Newton-Euler recursive scheme This numerical scheme is based on two loops to compute the vector of generalized efforts $\boldsymbol{\tau}_\mathrm{r}$. A kinetic loop propagates the twists starting from the arm base, while a second one transmits the efforts starting from the end-effector with a potential payload dynamics. The complete algorithm is given into details with the function $\texttt{InvDynRig}\left(\mathbf{q}_\mathrm{r}, \dot{\mathbf{q}}_\mathrm{r}, \ddot{\mathbf{q}}_\mathrm{r}\right)$ in Algorithm 4.1.

A first outward kinematic loop (i.e., from the base to the end-effector) computes the segment twists and accelerations using (4.28) and (4.36). The initial conditions are set for a fixed base, such that:

$$\mathbf{t}_0 = \mathbf{0}_{6\times1} \qquad\qquad \dot{\mathbf{t}}_0 = \mathbf{0}_{6\times1} \tag{4.46}$$

Then a second inward dynamic loop (i.e., from the end-effector to the base) computes the wrenches with (4.39), and the corresponding generalized efforts τ_i with (4.42). This loop starts from the external efforts applied **on** the end-effector with $\boldsymbol{w}_\mathrm{E}$:

$$\boldsymbol{w}_{n+1} = -\boldsymbol{w}_\mathrm{E} \tag{4.47}$$

If a payload is considered instead, the inward dynamic loop is initialized by its dynamics. Denoting $\boldsymbol{\omega}_{n+1}$, $\mathbf{t}_{n+1}$, and $\mathbf{M}_{\mathrm{r}_{n+1}}$, respectively, its angular rate, twist, and mass matrix, the wrench $\boldsymbol{w}_{n+1}$ influences the payload dynamics as follows using (4.20):

$$\boldsymbol{w}_{n+1} = \mathbf{M}_{\mathrm{r}_{n+1}}\,\dot{\mathbf{t}}_{n+1} + \boldsymbol{\Omega}_{n+1}\,\mathbf{M}_{\mathrm{r}_{n+1}}\,\mathbf{E}_v\,\mathbf{t}_{n+1} \tag{4.48}$$

It is worth noticing that the payload is actually considered as the $(n+1)^{th}$ body of the arm, driven by the wrench applied by the n^{th} segment. Matrices $\mathbf{A}_{n+1,n}$ and $\mathbf{P}_{n+1}$ are thus introduced to extend the algorithm in a simple way. $\mathbf{A}_{n+1,n}$ is defined by (4.29a) using the kinematic data of the tool described in (4.26). $\mathbf{P}_{n+1}$ and the corresponding coordinates $(q_{n+1}, \dot{q}_{n+1}, \ddot{q}_{n+1})$ have no physical meaning here, since the payload is assumed to be rigidly fixed at the end-effector. Nevertheless, they are introduced in the algorithm to keep the algorithm complexity as low as possible. Indeed, the payload is accounted for by only extending the kinematic loop with $i = n+1$, instead of adding extra computations.

Using this inverse dynamics function `InvDynRig`, M. W. Walker and D. E. Orin showed how the main terms of the closed-form equation in (4.45) can be computed (Walker and Orin, 1982). As it will be seen in the next section, the vector $\mathbf{h}_\mathrm{r}$ is quadratic in $\dot{\mathbf{q}}_\mathrm{r}$. It implies that

the mass matrix $\mathbf{D}_\mathrm{r}$ can be computed column by column using the following algorithm:

$$\mathbf{D}_\mathrm{r}(:,i) = \texttt{InvDynRig}(\mathbf{q}_\mathrm{r}, \mathbf{0}_{n\times 1}, \mathbf{e}_i) \quad , \quad \forall i = 1\dots n \tag{4.49}$$

where $\mathbf{e}_i$ is the i^{th} basis vector defined by $\forall j,\ \mathbf{e}_i(j) = \delta_{ij}$, with δ_{ij} the Kronecker delta. In the same way, $\mathbf{h}_\mathrm{r}$ is easily obtained by setting the acceleration to zero, since $\ddot{\mathbf{q}}_\mathrm{r}$ is only multiplying $\mathbf{D}_\mathrm{r}$:

$$\mathbf{h}_\mathrm{r}(\mathbf{q}_\mathrm{r}, \dot{\mathbf{q}}_\mathrm{r}) = \texttt{InvDynRig}(\mathbf{q}_\mathrm{r}, \dot{\mathbf{q}}_\mathrm{r}, \mathbf{0}_{n\times 1}) \tag{4.50}$$

This last expression will be used in forward dynamics to compute $\mathbf{h}_\mathrm{r}$ in an efficient manner.

Closed-form dynamics The closed-form equation of the arm dynamics is now derived based on the DeNOC matrices, and will be useful for control purpose. The expressions of the mass matrix and of the Coriolis and centrifugal vector are derived based on (Saha, 1999; Saha et al., 2013; Mohan and Saha, 2007).

Gathering together all the constrained dynamic models in (4.39), the system of equations governing the manipulator dynamics is obtained as follows:

$$\left\{\begin{array}{rcrcl}
\mathbf{M}_{\mathrm{r}_1}\dot{\mathbf{t}}_1 & + & \boldsymbol{\gamma}_{\mathrm{r}_1} & = & \boldsymbol{w}_1 - \mathbf{A}_{2,1}^\top \boldsymbol{w}_2 \\
\mathbf{M}_{\mathrm{r}_2}\dot{\mathbf{t}}_2 & + & \boldsymbol{\gamma}_{\mathrm{r}_2} & = & \boldsymbol{w}_2 - \mathbf{A}_{3,2}^\top \boldsymbol{w}_3 \\
 & & \vdots & & \\
\mathbf{M}_{\mathrm{r}_n}\dot{\mathbf{t}}_n & + & \boldsymbol{\gamma}_{\mathrm{r}_n} & = & \boldsymbol{w}_n - \mathbf{A}_{n+1,n}^\top \boldsymbol{w}_{n+1} \\
\hline
\mathbf{M}_{\mathrm{r}_{n+1}}\dot{\mathbf{t}}_{n+1} & + & \boldsymbol{\gamma}_{\mathrm{r}_{n+1}} & = & \boldsymbol{w}_{n+1}
\end{array}\right.$$

This system can be written in a more compact form using the global twist $\mathbf{t}$ and the global coordinate $\mathbf{q}_\mathrm{r}$ introduced for the Jacobian computation, as well as the global wrench $\boldsymbol{w} = [\,\boldsymbol{w}_1^\top, \dots, \boldsymbol{w}_n^\top\,]^\top$. The following block diagonal matrices are also introduced :

$$[\mathbf{M}_\mathrm{r}] = \mathrm{diag}(\mathbf{M}_{\mathrm{r}_i},\ i = 1\dots n)$$
$$[\boldsymbol{\Omega}] = \mathrm{diag}(\boldsymbol{\Omega}_i,\ i = 1\dots n)$$
$$[\mathbf{E}_v] = \mathrm{diag}(\mathbf{E}_v,\ i = 1\dots n)$$

Algorithm 4.1: Newton-Euler algorithm for rigid inverse dynamics (`InvDynRig`)

Function : $\boldsymbol{\tau}_{\mathrm{r}} = \texttt{InvDynRig}(\mathbf{q}_{\mathrm{r}},\ \dot{\mathbf{q}}_{\mathrm{r}},\ \ddot{\mathbf{q}}_{\mathrm{r}})$

Input : $\mathbf{q}_{\mathrm{r}},\ \dot{\mathbf{q}}_{\mathrm{r}},\ \ddot{\mathbf{q}}_{\mathrm{r}}$

Output : $\boldsymbol{\tau}_{\mathrm{r}}$

Data : $\left\{ {}^{i-1}R^{\mathrm{r}}_{i}\ ,\ \mathbf{A}^{\mathrm{r}}_{i,i-1}\ ,\ \mathbf{P}_i\ ,\ \mathbf{M}_{\mathrm{r}_i}\ ;\ i = 1 \dots n+1 \right\}$

`Initialize` $\left(\mathbf{t}_0 = \dot{\mathbf{t}}_0 = \mathbf{0}_{6\times 1},\ \mathbf{A}^{\mathrm{r}}_{1,0} = \mathbf{I}_6,\ \dot{\mathbf{A}}^{\mathrm{r}}_{1,0} = \mathbf{0}_{6\times 6}\right)$ // **KINETICS LOOP**

for $i = 1 \dots n+1$ **do** // Recursive computation of $\mathbf{t}_i$, $\dot{\mathbf{t}}_i$, and $\boldsymbol{\Omega}_i$

  if $i \leq n$ **then**

    ${}^{i}R_{i-1} = {}^{i-1}R^{\mathrm{r}}_{i}(q_i)^{\top}$ (4.22)

  else

    $\dot{q}_{n+1} = \ddot{q}_{n+1} = 0$ // Payload kinetics

    ${}^{i}R_{i-1} = \left({}^{n}R^{\mathrm{r}}_{n+1}\right)^{\top}$

  end

  ${}^{i}\boldsymbol{R}_{i-1} = \mathrm{diag}({}^{i}R_{i-1}, {}^{i}R_{i-1})$ (4.31)

  $\mathbf{A}_{i,i-1} = {}^{i}\boldsymbol{R}_{i-1}\ \mathbf{A}^{\mathrm{r}}_{i,i-1}$ (4.30a)

  $\dot{\mathbf{A}}_{i,i-1} = {}^{i}\boldsymbol{R}_{i-1}\ \dot{\mathbf{A}}^{\mathrm{r}}_{i,i-1}$ (4.38a)

  $\mathbf{t}_i = \mathbf{A}_{i,i-1}\,\mathbf{t}_{i-1} + \mathbf{P}_i\,\dot{q}_i$ (4.28)

  $\boldsymbol{\omega}_i = \mathbf{t}_i(4:6)$ (4.5)

  $\boldsymbol{\Omega}_i = \mathrm{diag}(\boldsymbol{\omega}_i^{\times},\ \boldsymbol{\omega}_i^{\times})$

  $\dot{\mathbf{t}}_i = \mathbf{A}_{i,i-1}\,\dot{\mathbf{t}}_{i-1} + \dot{\mathbf{A}}_{i,i-1}\,\mathbf{t}_{i-1} + \mathbf{P}_i\,\ddot{q}_i + \boldsymbol{\Omega}_i\,\mathbf{P}_i\,\dot{q}_i$ (4.36)

  if $i \leq n$ **then**

    $\dot{\mathbf{A}}^{\mathrm{r}}_{i+1,i} = \boldsymbol{\Omega}_i\,\mathbf{A}^{\mathrm{r}}_{i+1,i} - \mathbf{A}^{\mathrm{r}}_{i+1,i}\,\boldsymbol{\Omega}_i$ (4.37a)

  end

end

`Initialize` $\left(\boldsymbol{w}_{n+1} = \mathbf{M}_{\mathrm{r}_{n+1}}\,\dot{\mathbf{t}}_{n+1} + \left(\boldsymbol{\Omega}_{n+1}\mathbf{M}_{\mathrm{r}_{n+1}}\mathbf{E}_v\right)\mathbf{t}_{n+1}\right)$ // **DYNAMICS LOOP**

for $i = n \dots 1$ **do** // Recursive computation of $\boldsymbol{w}_i$, τ_i

  $\boldsymbol{\gamma}_{\mathrm{r}_i} = \boldsymbol{\Omega}_i\,\mathbf{M}_{\mathrm{r}_i}\,\mathbf{E}_v\,\mathbf{t}_i$ (4.40)

  $\boldsymbol{W}_i = \mathbf{M}_{\mathrm{r}_i}\dot{\mathbf{t}}_i + \boldsymbol{\gamma}_{\mathrm{r}_i}$ (4.39)

  $\boldsymbol{w}_i = \boldsymbol{W}_i + \mathbf{A}^{\top}_{i+1,i}\boldsymbol{w}_{i+1}$

  $\tau_i = \mathbf{P}_i^{\top}\boldsymbol{w}_i$ (4.42)

end

return $\boldsymbol{\tau}_{\mathrm{r}}$

to summarize the previous set of equations as follows, recalling that $\boldsymbol{w}_{n+1} = -\boldsymbol{w}_{\mathrm{E}}$:

$$[\mathbf{M}_{\mathrm{r}}]\,\dot{\mathbf{t}} + [\boldsymbol{\Omega}]\,[\mathbf{M}_{\mathrm{r}}]\,[\mathbf{E}_v]\,\mathbf{t} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{A}_{2,1} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{A}_{3,2} & \mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{A}_{n,n-1} & \mathbf{I} \end{bmatrix}^{\top}}_{=(\mathbf{I}-\mathbf{A})^{\top}=\left(\mathbf{N}_l^{-1}\right)^{\top}} \boldsymbol{w} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{A}_{n+1,n}^{\top} \end{bmatrix}}_{\triangleq \overline{\mathbf{A}}_{n+1,n}^{\top}} \boldsymbol{w}_{\mathrm{E}}$$

Reminding from (4.32) that $\mathbf{N}_l$ is the inverse of the matrix multiplying $\boldsymbol{w}$, the free-body Newton-Euler equations of all the segments are finally given by:

$$[\mathbf{M}_{\mathrm{r}}]\,\dot{\mathbf{t}} + [\boldsymbol{\Omega}]\,[\mathbf{M}_{\mathrm{r}}]\,[\mathbf{E}_v]\,\mathbf{t} = \mathbf{N}_l^{-\top}\,\boldsymbol{w} + \overline{\mathbf{A}}_{n+1,n}^{\top}\,\boldsymbol{w}_{\mathrm{E}} \tag{4.51}$$

In this equation, the twist $\mathbf{t}$ is a set of $6n$ coordinates, constrained by the kinetic relation $\mathbf{t} = \mathbf{N}_l\mathbf{N}_d\dot{\mathbf{q}}_{\mathrm{r}}$ in (4.32). To obtain the closed-form equation, the dimension of the previous set of equations must be reduced to the number of DoF. Namely, it is described by one equation per generalized coordinate in $\mathbf{q}_{\mathrm{r}}$, since they are, by definition, independent. The manipulator dynamics is now derived by reducing this number of equations from $6n$ to n by applying the kinematic constraints on $\mathbf{t}$.

To do so, the internal efforts are canceled by pre-multiplying the free-body equations in (4.51) by the transpose of the DeNOC matrix $\mathbf{N} = \mathbf{N}_l\,\mathbf{N}_d$. This method presented in (Mohan and Saha, 2007) is inspired from the Kane's formalism (Parsa, 2007). The twist and its time derivative are first developed using $\mathbf{N}$:

$$\mathbf{t} = \mathbf{N}\,\dot{\mathbf{q}}_{\mathrm{r}} \quad \Longrightarrow \quad \dot{\mathbf{t}} = \mathbf{N}\,\ddot{\mathbf{q}}_{\mathrm{r}} + \dot{\mathbf{N}}\,\dot{\mathbf{q}}_{\mathrm{r}}$$

The decoupled form of $\mathbf{N}$ is used to compute its time derivative in the inertial frame:

$$\begin{aligned} \dot{\mathbf{N}} &= \dot{\mathbf{N}}_l\mathbf{N}_d + \mathbf{N}_l\dot{\mathbf{N}}_d \\ &= \left(\mathbf{N}_l\dot{\mathbf{A}}\mathbf{N}_l\right)\mathbf{N}_d + \mathbf{N}_l\hat{\boldsymbol{\Omega}}\mathbf{N}_d \end{aligned} \tag{4.52}$$

where the time derivative of $\mathbf{N}_d = \mathrm{diag}(\mathbf{P}_i,\, i = 1 \ldots n)$ is straightforward using (4.38b),

and a the original definition of $\mathbf{N}_l$ is used to compute $\dot{\mathbf{N}}_l$[11].Recalling that the expression of the global twist-propagation matrix $\mathbf{A}$ is given by: $\mathbf{N}_l = (\mathbf{I} - \mathbf{A})^{-1}$, the time derivative is explicitly given using (4.38a):

$$\dot{\mathbf{N}}_l = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \dot{\mathbf{A}}_{2,1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \dot{\mathbf{A}}_{3,1} & \dot{\mathbf{A}}_{3,2} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \dot{\mathbf{A}}_{n,1} & \dot{\mathbf{A}}_{n,2} & \cdots & \dot{\mathbf{A}}_{n,n-1} & \mathbf{0} \end{bmatrix} = \mathbf{N}_l \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \dot{\mathbf{A}}_{2,1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \dot{\mathbf{A}}_{3,2} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \dot{\mathbf{A}}_{n,n-1} & \mathbf{0} \end{bmatrix}}_{\dot{\mathbf{A}}} \mathbf{N}_l \tag{4.53}$$

Replacing the twist and its time derivative in (4.51), and pre-multiplying by the DeNOC matrix $\mathbf{N}$, it yields

$$\mathbf{N}^\top [\mathbf{M}_\mathrm{r}] \left(\mathbf{N}\ddot{\mathbf{q}}_\mathrm{r} + \dot{\mathbf{N}}\dot{\mathbf{q}}_\mathrm{r} \right) + \mathbf{N}^\top [\boldsymbol{\Omega}] [\mathbf{M}_\mathrm{r}] [\mathbf{E}_v] \mathbf{N}\dot{\mathbf{q}}_\mathrm{r} = \mathbf{N}^\top \mathbf{N}_l^{-\top} \boldsymbol{w} + \mathbf{N}^\top \overline{\mathbf{A}}_{n+1,n}^\top \boldsymbol{w}_\mathrm{E}$$

On the RHS, $\mathbf{N}^\top \mathbf{N}_l^{-\top} \boldsymbol{w}$ reduces to $\mathbf{N}_d^\top \mathbf{N}_l^\top \mathbf{N}_l^{-\top} \boldsymbol{w} = \mathbf{N}_d^\top \boldsymbol{w} = \boldsymbol{\tau}_\mathrm{r}$, which is coherent with the fact that internal efforts do not work in (4.42) (Angeles and Lee, 1988; Saha et al., 2013). Furthermore, the computation of $\overline{\mathbf{A}}_{n+1,n}\, \mathbf{N}$ brings out the end-effector Jacobian $\mathbf{J}_\mathrm{E}$ given in (4.33).

Rigid dynamics of a robotic manipulator

The set of free-body equations in (4.51) is finally reduced to the manipulator dynamics under the classic closed form :

$$\mathbf{D}_\mathrm{r}(\mathbf{q}_\mathrm{r})\, \ddot{\mathbf{q}}_\mathrm{r} + \mathbf{C}_\mathrm{r}(\mathbf{q}_\mathrm{r}, \dot{\mathbf{q}}_\mathrm{r})\, \dot{\mathbf{q}}_\mathrm{r} = \boldsymbol{\tau}_\mathrm{r} + \mathbf{J}_\mathrm{E}^\top \boldsymbol{w}_\mathrm{E} \tag{4.54}$$

with

$$\mathbf{D}_\mathrm{r}(\mathbf{q}_\mathrm{r}) = \mathbf{N}^\top [\mathbf{M}_\mathrm{r}]\, \mathbf{N} \tag{4.55}$$

$$\mathbf{C}_\mathrm{r}(\mathbf{q}_\mathrm{r}, \dot{\mathbf{q}}_\mathrm{r}) = \mathbf{N}^\top [\mathbf{M}_\mathrm{r}] \dot{\mathbf{N}} + \mathbf{N}^\top [\boldsymbol{\Omega}] [\mathbf{M}_\mathrm{r}] [\mathbf{E}_v]\, \mathbf{N} \tag{4.56}$$

$$\boldsymbol{\tau}_\mathrm{r} = \mathbf{N}_d^\top \boldsymbol{w} \tag{4.57}$$

[11] $\dot{\mathbf{N}}_l$ is obtained by coming back to its initial expression in (4.32). It :

$$\mathbf{N}_l (\mathbf{I} - \mathbf{A}) = \mathbf{I} \implies \dot{\mathbf{N}}_l (\mathbf{I} - \mathbf{A}) - \mathbf{N}_l \dot{\mathbf{A}} = \mathbf{0} \implies \dot{\mathbf{N}}_l \mathbf{N}_l^{-1} = \mathbf{N}_l \dot{\mathbf{A}} \implies \dot{\mathbf{N}}_l = \mathbf{N}_l \dot{\mathbf{A}} \mathbf{N}_l$$

The previous model is given under an analytical form which suits well for global analysis and control. On the other hand, dedicated algorithms are available for numerical simulation, with recursive computations of the mass matrix and of the Coriolis and centrifugal vector. They are more efficient than the direct computation by using the matrices $[\mathbf{M}_\mathrm{r}]$, $\mathbf{N}$, $\dot{\mathbf{N}}$ and $[\boldsymbol{\Omega}]$, whose size can be relatively high. The details of these algorithms are recalled for the rigid case in Appendix A.3, based on (Saha, 1999; Saha et al., 2013; Mohan and Saha, 2007, 2009). They will be extended to the flexible case and to the moving base in the sequel.

N.B.: If a payload is to be considered, the wrench at the end-effector $\boldsymbol{w}_\mathrm{E}$ will act as a feedback on the dynamics, since it is defined by the last segment dynamics. Indeed, the payload twist is merely given by $\mathbf{t}_{n+1} = \mathbf{A}_{n+1,n}\mathbf{t}_n$ since the matrix $\mathbf{P}_{n+1}$ does not exist[a] in the absence of DoF. Taking into account its dynamics given in (4.48), the set of equations in (4.51) is re-written:

$$[\mathbf{M}_\mathrm{r}]\,\dot{\mathbf{t}} + [\boldsymbol{\Omega}]\,[\mathbf{M}_\mathrm{r}]\,[\mathbf{E}_v]\,\mathbf{t} = \mathbf{N}_l^{-\top}\,\boldsymbol{w} \tag{4.58}$$

where the twist $\mathbf{t}$, the wrench $\boldsymbol{w}$ and the diagonal matrices $[.]$ are augmented with their $(n+1)^{th}$ value; namely, the payload twist $\mathbf{t}_{n+1}$, its wrench $\boldsymbol{w}_{n+1}$, its mass matrix $\mathbf{M}_{\mathrm{r}_{n+1}}$, and the cross-product of its angular rate $\boldsymbol{\Omega}_{n+1}$.
Since $\mathbf{t}$ has one extra term, the constraints are re-written by only modifying the matrix $\mathbf{N}_l$ while the matrix $\mathbf{N}_d$ is left unchanged:

$$\left[\begin{array}{c} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{t}_3 \\ \vdots \\ \mathbf{t}_n \\ \hline \mathbf{t}_{n+1} \end{array}\right] = \underbrace{\left[\begin{array}{ccccc} \mathbf{I}_6 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{2,1} & \mathbf{I}_6 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{3,1} & \mathbf{A}_{3,2} & \mathbf{I}_6 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \cdots & \mathbf{0} \\ \mathbf{A}_{n,1} & \mathbf{A}_{n,2} & \cdots & \mathbf{A}_{n,n-1} & \mathbf{I}_6 \\ \hline \mathbf{A}_{n+1,1} & \mathbf{A}_{n+1,2} & \cdots & \mathbf{A}_{n+1,n-1} & \mathbf{A}_{n+1,n} \end{array}\right]}_{=\mathbf{N}_l} \underbrace{\left[\begin{array}{ccccc} \mathbf{P}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_3 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_n \end{array}\right]}_{=\mathbf{N}_d} \dot{\mathbf{q}}_\mathrm{r} \tag{4.59}$$

To summarize, the dynamic equations with a payload are exactly the same as (4.54) to (4.56) by canceling the wrench $\boldsymbol{w}_\mathrm{E}$, and using the augmented version of $\mathbf{N}_l$, $[\mathbf{M}_\mathrm{r}]$, $[\boldsymbol{\Omega}]$, and $[\mathbf{E}_v]$.

[a] Matrix $\mathbf{P}_{n+1}$ and coordinate q_{n+1} are only set to zero in the algorithms to provide simpler programs.

4.2.5 Forward Dynamics

Forward dynamics are the key components for dynamic simulation. It consists in computing the accelerations resulting from efforts applied at the joints by $\boldsymbol{\tau}_\mathrm{r}$, and assuming a given configuration of the manipulator $(\mathbf{q}_\mathrm{r}, \dot{\mathbf{q}}_\mathrm{r})$. With the closed-form equation, this task is actually performed by inverting the mass matrix:

$$\ddot{\mathbf{q}}_\mathrm{r} = \mathbf{D}_\mathrm{r}(\mathbf{q}_\mathrm{r})^{-1} \left(\boldsymbol{\tau}_\mathrm{r} - \mathbf{h}_\mathrm{r}(\mathbf{q}_\mathrm{r}, \dot{\mathbf{q}}_\mathrm{r})\right) \tag{4.60}$$

where the external efforts at end-effector have been canceled, by rather considering a free motion or a payload already included in the computation of $\mathbf{D}_\mathrm{r}$ and $\mathbf{h}_\mathrm{r}$.

To perform forward dynamics, Coriolis and centrifugal terms are assumed to be known first. Then the inversion of $\mathbf{D}_\mathrm{r}$ is performed by using its recursive structure described into details in Algorithm A.1. Instead of inverting it roughly and downgrading the numerical efficiency, an algorithm is proposed by S. K. Saha to apply a *Recursive Gaussian Elimination (RGE)* technique on the block structure of the mass matrix (Saha, 1997). An overview is given in the following, while the algorithm is thoroughly covered and demonstrated in Appendix A.3.4.

This approach is based on the $\mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top$ decomposition of the symmetric mass matrix $\mathbf{D}_\mathrm{r}$:

$$\exists \mathbf{U} \in \mathbb{R}^{n\times n},\ \boldsymbol{\Delta} \in \mathbb{R}^{n\times n}, \quad \mathbf{D}_\mathrm{r} = \mathbf{U}\,\boldsymbol{\Delta}\,\mathbf{U}^\top$$

with $\mathbf{U}$ a block upper triangular matrix and $\boldsymbol{\Delta}$ a block diagonal matrix.

$$\mathbf{U} = \begin{bmatrix} \mathbf{I} & \mathbf{U}_{1,2} & \cdots & \mathbf{U}_{1,n} \\ \mathbf{0} & \mathbf{I} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{U}_{n-1,n} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{bmatrix} \quad \boldsymbol{\Delta} = \begin{bmatrix} \boldsymbol{\Delta}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Delta}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\Delta}_n \end{bmatrix} \quad \mathbf{U}^\top = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{U}_{1,2}^\top & \mathbf{I} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{U}_{1,n}^\top & \cdots & \mathbf{U}_{n-1,n}^\top & \mathbf{I} \end{bmatrix}$$

These terms are obtained analytically as functions of the twist-propagation and joint-rate-propagation matrices in Appendix A.3. These expressions are necessary in order to solve the system $\left(\mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top\right)\mathbf{X} = \mathbf{b}$, with $\mathbf{X}$ unknown. Three steps are performed:

1. Solve: $\mathbf{U}\widehat{\mathbf{X}} = \mathbf{b}$
2. Solve: $\boldsymbol{\Delta}\bar{\mathbf{X}} = \widehat{\mathbf{X}}$ $(= \mathbf{U}^{-1}\,\mathbf{b})$
3. Solve: $\mathbf{U}^\top\mathbf{X} = \bar{\mathbf{X}}$ $(= \boldsymbol{\Delta}^{-1}\,\mathbf{U}^{-1}\,\mathbf{b})$

Based on the recursive solution of these three sub-systems of equations, the whole forward

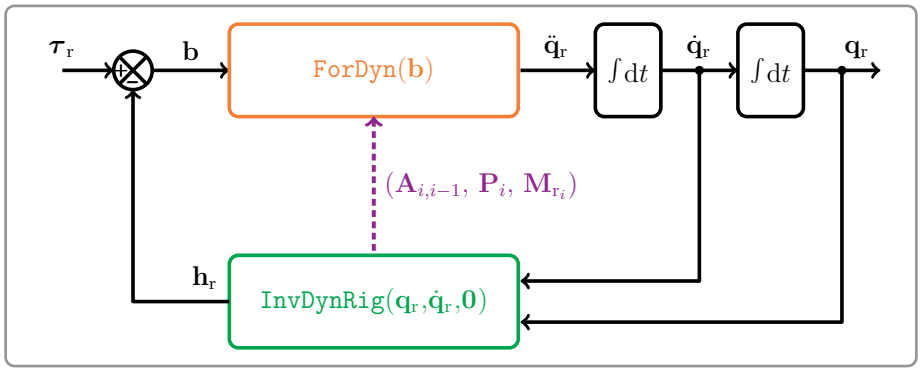


Figure 4.3: Forward dynamics scheme for numerical simulation of rigid multi-body systems

dynamics algorithm is given in Algorithm A.3. The function is called `ForDyn`, and is valid for both rigid or flexible systems, provided that the twist-propagation and joint-rate-propagation matrices are updated accordingly.

Finally, a general scheme is proposed in Figure 4.3 to illustrate the use of the previous algorithms to perform the simulation of a rigid multi-body system. It is supposed that the inputs are the joint torques $\boldsymbol{\tau}_{\mathrm{r}}$, and the output the joint angles $\mathbf{q}_{\mathrm{r}}$.

CHAPTER 5 FLEXIBLE FIXED-BASE ROBOT MODELING

The flexible behavior of the segments is now investigated to be accounted for in the global manipulator dynamics. The main extension lies in the dynamic model of a single flexible segment, which is thoroughly derived in the scope of the AMM. Kinematics and kinetics of the robotic arm are also extended to encompass the flexible deflections in translation and rotation. In the following, the dynamic model of a single flexible segment is derived first, and then the previous modeling algorithms are adapted for chain-like manipulator made up of these flexible bodies.

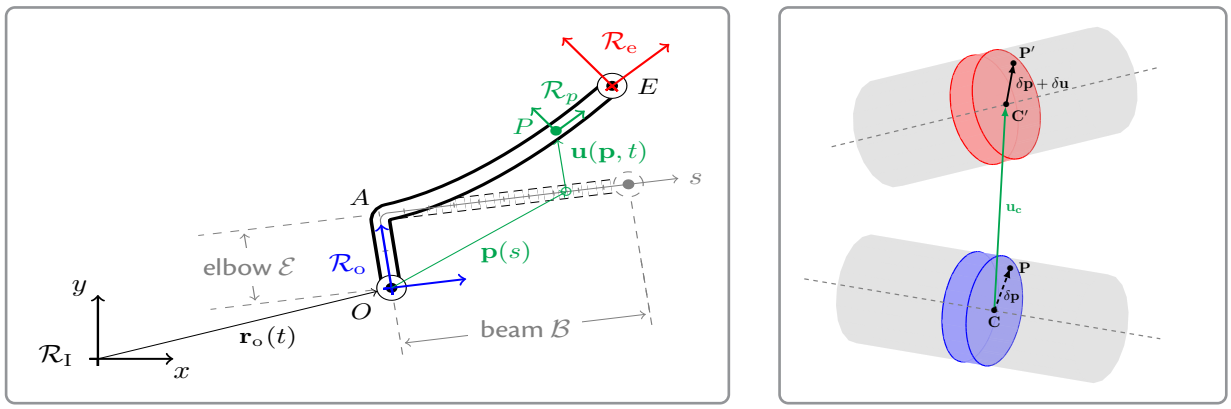
5.1 Model of a Single Flexible Segment

As for the rigid case, the same path through kinematics and kinetics is followed to obtain the dynamic model of single segment with distributed flexibility. This model may also be used to include the in-joint elasticity as well, by considering a rotor body subject to torsion. In the sequel, kinematics is augmented to include the linear/angular deflections along the segment, then kinetics are described by additional terms leading to the corrective mass matrices in the kinetic energy expression. Using this latter, the quasi-Lagrangian equations are still used to derived the flexible dynamic model of a single segment constrained at both ends.

Among the various refinements existing to model a flexible body, Euler-Bernoulli beam theory is adopted here, based on the hypothesis of slender segments. The approach developed in the following could be adapted for different models, as long as separability in time and space holds to express the segment kinematics and kinetics. With this assumption, kinetic and potential energies are derived by differentiating kinematics, and the corresponding Lagrangian is used to derive the flexible segment dynamics using the hybrid state equations.

5.1.1 Kinematics

As mentioned earlier, the segment is assumed to have an elbow shape, such that a short rigid part is considered at the beginning and a flexible slender one for the remaining. Figure 4.1 for rigid case is updated in Figure 5.1 with the flexible displacement along the beam. The position and attitude of the two body-fixed frames $\mathcal{R}_{\mathrm{o}}$ and $\mathcal{R}_{\mathrm{e}}$ are shifted from the rigid case, due to the flexible motion at the end-tip. The frame $\mathcal{R}_{\mathrm{o}}$ meets the notion of *floating frame* introduced by A. A. Shabana to represent the frame that is rigidly fixed to the undeformed segment (Shabana, 1997).



(a) (b)

Figure 5.1: Kinematics of a flexible segment; (a) Frames and vectors describing the global motion through the reference point O; (b) Linear and angular deflection of the relative position $\delta\mathbf{p}$ of any point P in a beam section, transforming (C, P) in the undeformed state into (C', P') (N.B. Vectors are still expressed in $\mathcal{R}_o$).

Along the segment, any section experiences a linear and angular deformation in the three directions, modeled by traction, bending and torsion, as illustrated in Figure 5.2. Using the usual convention of beam theory (X-axis along the elongated section), the considered flexible deformations are:

- **traction**: X-translation only (T_x);
- **Y-bending**: Y-translation and Z-rotation (T_y, R_z);
- **Z-bending**: Z-translation and $(-Y)$-rotation (T_z, R_{-y});
- **torsion**: X-rotation only (R_x).

At the rotational level, the flexible angular drift of the cross-section is described by a set of Euler angles. The attitude is described by the frame $\mathcal{R}_p$ which is rigidly fixed at point P, at the curvilinear abscissa s. Its relative attitude $\boldsymbol{\psi}(s,t)$ w.r.t. the base frame $\mathcal{R}_o$ is only due to the flexible motion. It is denoted by:

$$\boldsymbol{\psi} = \begin{bmatrix} \psi_x & \psi_y & \psi_z \end{bmatrix}^\top \tag{5.1}$$

where (ψ_x, ψ_y, ψ_z) describe, respectively, the angular deflection due to torsion, Z-bending and Y-bending. The inertial attitude of $\mathcal{R}_p$ is then obtained by introducing the additional rotation ${}^oR_p$ corresponding to these Euler angles:

$${}^IR_p = {}^IR_o\,{}^oR_p(\boldsymbol{\psi}) \tag{5.2}$$

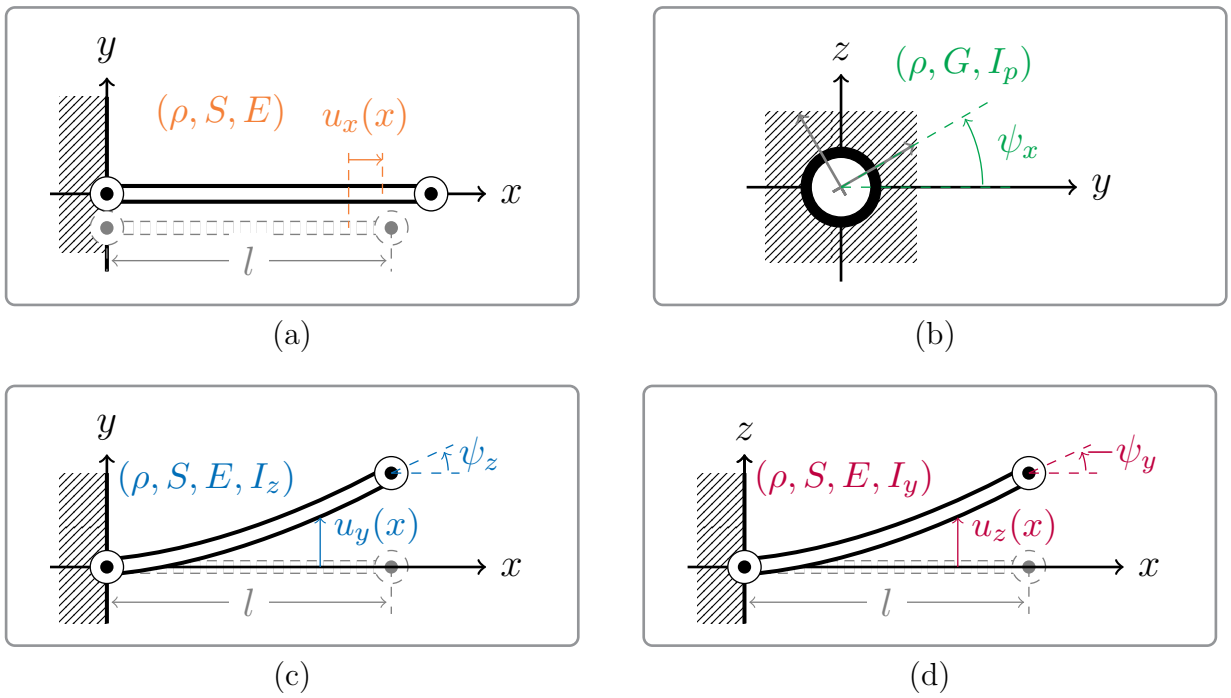


Figure 5.2: Flexible behavior of a beam in each direction; (a) Traction along X axis; (b) Torsion around X axis; (c) Bending in (X, Y) plane; (d) Bending in (X, Z) plane; where ρ is the material density, S the beam cross-section, E the material Young modulus, G the material Poisson modulus, I_z (resp. I_y) the second moment of area for y-bending (resp. z-bending) and I_p the polar moment of area.

As advised in (Damaren and Sharf, 1995) for enhanced numerical stability, the 3-2-1 Euler convention given explicitly in (4.2) is used here, such that:

$$\begin{aligned} {}^{\mathrm{o}}R_{\mathrm{e}}(s,t) &= R_{\text{3-2-1}}(\boldsymbol{\psi}_{\mathrm{e}}(s,t)) \\ &= \mathrm{Rot}(\psi_{\mathrm{e}_z}, Z)\,\mathrm{Rot}(\psi_{\mathrm{e}_y}, Y)\,\mathrm{Rot}(\psi_{\mathrm{e}_x}, X) \end{aligned} \tag{5.3}$$

For the linear deflection, the inertial position $\mathbf{r}(\mathbf{p}, t)$ of any point P along the segment consists of three main terms, illustrated in Figure 5.1:

$$\mathbf{r}(\mathbf{p}, t) = \mathbf{r}_{\mathrm{o}}(t) + \Big(\mathbf{p} + \mathbf{u}(\mathbf{p}, t)\Big) \tag{5.4}$$

with $\mathbf{u}$ the flexible deformation of P, expressed in $\mathcal{R}_{\mathrm{o}}$.

As mentioned for the rigid case, the undeformed position $\mathbf{p}$ is further decomposed between the longitudinal coordinate along the centroidal axis with $\mathbf{p}_c$, and the transverse one perpen-

dicular to this axis with $\delta\mathbf{p}$:

$$\mathbf{p} = \mathbf{p}_c(s) + \delta\mathbf{p}(y,z) \tag{5.5}$$

Hence, the flexible deformation of a point P also decomposes into the centroidal axis deflection $\mathbf{u}_c$ and the transverse one $\delta\mathbf{u}$. This latter is due to the rotation of the cross-section, which is supposed to stay planar. It reads as follows:

$$\mathbf{u}(\mathbf{p},t) = \mathbf{u}_c(s,t) + \delta\mathbf{u}(\mathbf{p},t) \tag{5.6}$$

Flexible coordinates The previous flexible quantities $(\boldsymbol{\psi}, \mathbf{u}_c, \delta\mathbf{u})$ in (5.1) and (5.6) are developed based on the AMM. The Euler-Bernoulli beam models are given by PDEs related to each flexible directions (i.e., traction, torsion, bending) in Appendix B.1. It yields a linear model of flexibility with a separability in time and space (De Luca and Siciliano, 1991). Both flexible translation and rotation are written as the summation of n_f modes, including n_x modes in traction, $n_{y/z}$ modes in y/z-bending, and n_α modes in torsion. The computation of the clamped-loaded modes are given in detail in Appendix B.1, and their shapes are denoted $\boldsymbol{\phi}_{x/y/z/\alpha}$. A vector of time amplitudes $\boldsymbol{\delta}_{x/y/z/\alpha}$ is associated to each of these spatial modes. The flexible displacement and rotation are then written in a compact form by:

$$\mathbf{u}_c(s,t) = \boldsymbol{\Phi}(s)\,\boldsymbol{\delta}(t) \tag{5.7a}$$

$$\boldsymbol{\psi}(s,t) = \boldsymbol{\Delta}(s)\,\boldsymbol{\delta}(t) \tag{5.7b}$$

$$\delta\mathbf{u}(\mathbf{p},t) = \left(\boldsymbol{\Delta}(s)\,\boldsymbol{\delta}(t)\right)^{\times}\delta\mathbf{p}(y,z) \tag{5.7c}$$

where the time amplitudes are merged in $\boldsymbol{\delta} \in \mathbb{R}^{n_f}$, called the *flexible coordinates*, and the mode shapes for translational and rotational displacements are stored, respectively, in $\boldsymbol{\Phi} \in \mathbb{R}^{3\times n_f}$ and $\boldsymbol{\Delta} \in \mathbb{R}^{3\times n_f}$. Their contents is derived explicitly in Appendix B.1. Their expression read as follows:

$$\boldsymbol{\delta}(s) = \begin{bmatrix} \boldsymbol{\delta}_x^\top & \boldsymbol{\delta}_y^\top & \boldsymbol{\delta}_z^\top & \boldsymbol{\delta}_\alpha^\top \end{bmatrix}^\top$$

$$\boldsymbol{\Phi}(s) = \begin{bmatrix} \boldsymbol{\phi}_x^\top(s) & \mathbf{0}_{1\times n_y} & \mathbf{0}_{1\times n_z} & \mathbf{0}_{1\times n_\alpha} \\ \mathbf{0}_{1\times n_x} & \boldsymbol{\phi}_y^\top(s) & \mathbf{0}_{1\times n_z} & \mathbf{0}_{1\times n_\alpha} \\ \mathbf{0}_{1\times n_x} & \mathbf{0}_{1\times n_y} & \boldsymbol{\phi}_z^\top(s) & \mathbf{0}_{1\times n_\alpha} \end{bmatrix}$$

$$\boldsymbol{\Delta}(s) = \begin{bmatrix} \mathbf{0}_{1\times n_x} & \mathbf{0}_{1\times n_y} & \mathbf{0}_{1\times n_z} & \boldsymbol{\phi}_\alpha^\top(s) \\ \mathbf{0}_{1\times n_x} & \mathbf{0}_{1\times n_y} & -\boldsymbol{\phi}_z'^\top(s) & \mathbf{0}_{1\times n_\alpha} \\ \mathbf{0}_{1\times n_x} & \boldsymbol{\phi}_y'^\top(s) & \mathbf{0}_{1\times n_z} & \mathbf{0}_{1\times n_\alpha} \end{bmatrix}$$

The unconstrained coordinates of the rigid body are augmented by the corresponding flexible coordinates $\boldsymbol{\delta}$. These latter are additional DoFs for the body, such that the final coordinates of a free flexible segment read:

$$\mathbf{x}_{\mathrm{o}} = \begin{bmatrix} \mathbf{r}_{\mathrm{o}}^{\top} & \boldsymbol{\Psi}_{\mathrm{o}}^{\top} & \boldsymbol{\delta}^{\top} \end{bmatrix}^{\top} \tag{5.8}$$

5.1.2 Kinetics

As the vector of unconstrained coordinates, the twist describing the motion and internal state of the segment is augmented with the derivative of these flexible coordinates (Mohan and Saha, 2009). The reference twist attached to the segment base is merely an extended version of (4.5):

$$\mathbf{t}_{\mathrm{o}} = \begin{bmatrix} \boldsymbol{v}_{\mathrm{o}} \\ \boldsymbol{\omega}_{\mathrm{o}} \\ \dot{\boldsymbol{\delta}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times n_f} \\ \mathbf{0}_{3\times 3} & R_{\boldsymbol{\Psi}_{\mathrm{o}}} & \mathbf{0}_{3\times n_f} \\ \mathbf{0}_{n_f\times 3} & \mathbf{0}_{n_f\times 3} & \mathbf{I}_{n_f} \end{bmatrix} \dot{\mathbf{x}}_{\mathrm{o}} \tag{5.9}$$

With the previous relations giving the flexible position along the segment, the linear velocity of any point P is readily updated with the AMM formalism. It is worth noting that $\mathbf{u}$ is the only time-dependent vector w.r.t. $\mathcal{R}_{\mathrm{o}}$ in (5.4). Using in addition the small angle hypothesis, the angular rate of the beam section is also updated by the flexible deflections. Rigid equation (4.5) is thus updated by[1]:

$$\boldsymbol{v}(\mathbf{p},t) = \boldsymbol{v}_{\mathrm{o}}(t) + \frac{\partial \mathbf{u}}{\partial t}(\mathbf{p},t) + \boldsymbol{\omega}_{\mathrm{o}} \times \Big(\mathbf{p} + \mathbf{u}(\mathbf{p},t)\Big) \tag{5.10a}$$

$$\boldsymbol{\omega}(s,t) = \boldsymbol{\omega}_{\mathrm{o}}(t) + R_{\boldsymbol{\psi}} \, \frac{\partial \boldsymbol{\psi}}{\partial t}(s,t) \tag{5.10b}$$

where $(\boldsymbol{v}_{\mathrm{o}}, \boldsymbol{\omega}_{\mathrm{o}})$ and $(\boldsymbol{v}, \boldsymbol{\omega})$ are the inertial speed and angular rate of, respectively, frame $\mathcal{R}_{\mathrm{o}}$ at O and frame $\mathcal{R}_{\mathrm{p}}$ rigidly attached at P and rotating with its beam section. The term $R_{\boldsymbol{\psi}}$ is introduced in (4.5) to denote the transformation from the time derivative of Euler angles to the corresponding angular rate.

Using the AMM expressions in (5.7), and the flexible decompositions in (5.5) and (5.6), the

[1] All vectors in (5.4) are expressed in $\mathcal{R}_{\mathrm{o}}$ for sake of simplicity, so the time derivative w.r.t. $\mathcal{R}_{\mathrm{I}}$ brings out the angular rate $\boldsymbol{\omega}_{\mathrm{o}}$ and not the local one $\boldsymbol{\omega}$.

time derivatives are developed explicitly by:

$$\frac{\partial \mathbf{u}}{\partial t}(\mathbf{p},t) = \mathbf{\Phi}(s)\,\dot{\boldsymbol{\delta}}(t) - \delta\mathbf{p}(y,z)^{\times}\,\mathbf{\Delta}(s)\dot{\boldsymbol{\delta}}(t)$$
$$\frac{\partial \boldsymbol{\psi}}{\partial t}(s,t) = \mathbf{\Delta}(s)\,\dot{\boldsymbol{\delta}}(t)$$

The kinetic relation used to propagate the twist for the rigid case in (4.7) is now updated for the flexible case by:

$$\mathbf{t}_{\mathrm{p}}(\mathbf{p},t) = \underbrace{\left[\begin{array}{cc|c} \mathbf{I}_3 & -\left(\mathbf{p} + \left(\mathbf{\Phi} - \delta\mathbf{p}^{\times}\mathbf{\Delta}\right)\boldsymbol{\delta}\right)^{\times} & \left(\mathbf{\Phi} - \delta\mathbf{p}^{\times}\mathbf{\Delta}\right) \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & R_{\boldsymbol{\psi}}\,\mathbf{\Delta} \end{array}\right]}_{\triangleq \mathbf{A}_{\mathrm{p,o}}(\mathbf{p},t)} \mathbf{t}_{\mathrm{o}}(t) \tag{5.11}$$

where the twist propagation matrix $\mathbf{A}_{\mathrm{p,o}}$ is augmented from the rigid case (Mohan and Saha, 2009). When derived for the end-tip E, the transverse position and deflections cancel by assuming that this point is located on the centroidal axis. More specifically, $\delta\mathbf{p}(0,0) = \mathbf{0}_{3\times1}$ and so $\mathbf{u}_{\mathrm{e}} = \mathbf{\Phi}_{\mathrm{e}}\,\boldsymbol{\delta}$. The flexible end-tip twist is thus given by:

$$\mathbf{t}_{\mathrm{e}}(t) = \underbrace{\left[\begin{array}{cc|c} \mathbf{I}_3 & -\left(\mathbf{p}_{\mathrm{e}} + \mathbf{\Phi}_{\mathrm{e}}\,\boldsymbol{\delta}(t)\right)^{\times} & \mathbf{\Phi}_{\mathrm{e}} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & R_{\boldsymbol{\psi}_{\mathrm{e}}}\,\mathbf{\Delta}_{\mathrm{e}} \end{array}\right]}_{\triangleq \mathbf{A}_{\mathrm{e,o}}(t)} \mathbf{t}_{\mathrm{o}}(t) \tag{5.12}$$

5.1.3 Kinetic Energy

The derivation of the kinetic energy is more cumbersome for the flexible case because the linear speed of a point P involves more terms, and above all, it mixes many explicit dependencies upon the curvilinear abscissa s in (5.11). The global kinetic energy T is now expressed as a function of $\boldsymbol{v}$, given by the upper row of $\mathbf{A}_{\mathrm{p,o}}$ only. It is divided into three terms: one for the rigid contribution $\mathbf{v}_r$, one for the flexible translation $\mathbf{v}_u$, and one for the flexible rotation $\mathbf{v}_\psi$:

$$\boldsymbol{v} = \mathbf{v}_r + \mathbf{v}_u + \mathbf{v}_\psi \tag{5.13}$$

with

$$\mathbf{v}_r = \left[\begin{array}{ccc} \mathbf{I}_3 & -\,\mathbf{p}^{\times} & \mathbf{0}_{3\times n_f} \end{array}\right] \mathbf{t}_{\mathrm{o}} \tag{5.14a}$$
$$\mathbf{v}_u = \left[\begin{array}{ccc} \mathbf{0}_{3\times3} & -\left(\mathbf{\Phi}\,\boldsymbol{\delta}\right)^{\times} & \mathbf{\Phi} \end{array}\right] \mathbf{t}_{\mathrm{o}} \tag{5.14b}$$
$$\mathbf{v}_\psi = \left[\begin{array}{ccc} \mathbf{0}_{3\times3} & \left(\delta\mathbf{p}^{\times}\left(\mathbf{\Delta}\,\boldsymbol{\delta}\right)\right)^{\times} & -\delta\mathbf{p}^{\times}\mathbf{\Delta} \end{array}\right] \mathbf{t}_{\mathrm{o}} \tag{5.14c}$$

The differential energy $\frac{1}{2}\boldsymbol{v}^{\top}\boldsymbol{v}\mathrm{d}m$ is then integrated on the whole segment to express the global

flexible kinetic energy:

$$T = \frac{1}{2}\int_{\mathcal{L}} \left(\boldsymbol{v}^\top \boldsymbol{v}\right) \mathrm{d}m$$

The velocity $\boldsymbol{v}$ is expanded according to (5.14), and the hypothesis of an elbow-shaped segment allows to reduce the integration of the flexible matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Delta}$. As illustrated in Figure 5.2, the flexible speed contributions satisfy: $\mathbf{v}_u(\mathbf{p},t) = \mathbf{v}_\psi(\mathbf{p},t) = 0$, $\forall \mathbf{p} \in \mathcal{E}$ (i.e., along the rigid section). Therefore, T can be sorted by the rigid and flexible nature of its terms, as follows:

$$\begin{aligned} T = & \underbrace{\frac{1}{2}\int_{\mathcal{L}} \mathbf{v}_r^\top \mathbf{v}_r \mathrm{d}m}_{T_{\mathrm{rr}}} + \underbrace{\frac{1}{2}\int_{\mathcal{B}} \mathbf{v}_u^\top \mathbf{v}_u \mathrm{d}m}_{T_{\mathrm{uu}}} + \underbrace{\frac{1}{2}\int_{\mathcal{B}} \mathbf{v}_\psi^\top \mathbf{v}_\psi \mathrm{d}m}_{T_{\psi\psi}} \\ & + \underbrace{\int_{\mathcal{B}} \mathbf{v}_r^\top \mathbf{v}_u \mathrm{d}m}_{T_{\mathrm{ru}}+T_{\mathrm{ur}}} + \underbrace{\int_{\mathcal{B}} \mathbf{v}_r^\top \mathbf{v}_\psi \mathrm{d}m}_{T_{\mathrm{r}\psi}+T_{\psi\mathrm{r}}} + \underbrace{\int_{\mathcal{B}} \mathbf{v}_u^\top \mathbf{v}_\psi \mathrm{d}m}_{T_{\mathrm{u}\psi}+T_{\psi\mathrm{u}}} \end{aligned} \tag{5.15}$$

where each term is developed separately in Appendix B.2, in order to keep the distinction between rigid and flexible contributions. Once these computations are performed, the global kinetic energy of a flexible segment is set under matrix form by replacing expressions (B.18), (B.28), (B.32), (B.37), (B.41) and (B.44) of Appendix B.2, into (5.15). It yields:

$$T = \frac{1}{2}\mathbf{t}_{\mathrm{o}}^\top \underbrace{\left[\begin{array}{ccc|c} \mathbf{M}_{vv} & \mathbf{M}_{v\omega} & & \mathbf{M}_{v\delta} \\ * & \mathbf{M}_{\omega\omega} & & \mathbf{M}_{\omega\delta} \\ \hline & & & \\ * & * & & \mathbf{M}_{\delta\delta} \end{array}\right]}_{\mathbf{M}_{\mathrm{f}}} \mathbf{t}_{\mathrm{o}} \tag{5.16}$$

with:

$$\begin{aligned} \mathbf{M}_{vv} &= m\mathbf{I}_3 & \mathbf{M}_{\omega\omega} &= \mathbf{I}_{\mathrm{o}} + \mathbf{I}_{ru}(\boldsymbol{\delta}) + \mathbf{I}_{r\psi}(\boldsymbol{\delta}) + \mathbf{I}_{uu}(\boldsymbol{\delta},\boldsymbol{\delta}) + \mathbf{I}_{\psi\psi}(\boldsymbol{\delta},\boldsymbol{\delta}) \\ \mathbf{M}_{v\omega} &= -\left(\mathbf{c} + \mathbf{V}_u\boldsymbol{\delta}\right)^\times & \mathbf{M}_{\omega\delta} &= \mathbf{H}_{ru} + \mathbf{H}_{r\psi} + \mathbf{H}_{uu}(\boldsymbol{\delta}) + \mathbf{H}_{\psi\psi}(\boldsymbol{\delta}) \\ \mathbf{M}_{v\delta} &= \mathbf{P}_{ru} & \mathbf{M}_{\delta\delta} &= \bar{\mathbf{Z}}_{\boldsymbol{uu}} + \bar{\mathbf{Z}}_{\boldsymbol{\psi\psi}} \end{aligned}$$

where matrices $\mathbf{P}_{ii}$, $\mathbf{H}_{ii}$, and $\mathbf{I}_{ii}$, contribute to, respectively, the flexible expressions of the linear and angular momentum and to the inertia tensor. Their exact expression is detailed in Appendix B.2. The subscripts indicate their origin in regards to (5.15).

This analytical expression is similar to equation (24) in (Mohan and Saha, 2009), but the main contribution here is the exact computation of the integrals with the AMM representation, and their ranking according to the linear or quadratic dependency upon $\boldsymbol{\delta}$. Doing this, the constant terms can be computed and stored off-line, while the time-varying ones can be studied in simulation to investigate their impact. Denoting by $\mathbf{M}_{\mathrm{f}}^{(i)}$ the matrix containing the constant, linear and quadratic terms with $i \in \{0,1,2\}$, $\mathbf{M}_{\mathrm{f}}$ is sorted as follows:

$$\mathbf{M}_\mathrm{f} = \underbrace{\mathbf{M}_\mathrm{f}^{(0)}}_{\mathcal{O}(1)} + \underbrace{\mathbf{M}_\mathrm{f}^{(1)}}_{\mathcal{O}(\boldsymbol{\delta})} + \underbrace{\mathbf{M}_\mathrm{f}^{(2)}}_{\mathcal{O}(\boldsymbol{\delta}^2)} \tag{5.17}$$

with

$$\mathbf{M}_\mathrm{f}^{(0)} = \left[\begin{array}{ccc} m\mathbf{I}_3 & (-\mathbf{c})^\times & \mathbf{P}_{ru} \\ * & \mathbf{I}_O & \mathbf{H}_{ru} + \mathbf{H}_{r\psi} \\ \hline * & * & \bar{\mathbf{Z}}_{\boldsymbol{uu}} + \bar{\mathbf{Z}}_{\boldsymbol{\psi\psi}} \end{array}\right] \tag{5.18a}$$

$$\mathbf{M}_\mathrm{f}^{(1)}(\boldsymbol{\delta}) = \left[\begin{array}{ccc} \mathbf{0}_{3\times3} & (-\mathbf{V}_u\boldsymbol{\delta})^\times & \mathbf{0}_{3\times n_f} \\ * & \mathbf{I}_{ru}(\boldsymbol{\delta}) + \mathbf{I}_{r\psi}(\boldsymbol{\delta}) & \mathbf{H}_{uu}(\boldsymbol{\delta}) + \mathbf{H}_{\psi\psi}(\boldsymbol{\delta}) \\ \hline * & * & \mathbf{0}_{n_f\times n_f} \end{array}\right] \tag{5.18b}$$

$$\mathbf{M}_\mathrm{f}^{(2)}(\boldsymbol{\delta},\boldsymbol{\delta}) = \left[\begin{array}{ccc} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n_f} \\ * & \mathbf{I}_{uu}(\boldsymbol{\delta},\boldsymbol{\delta}) + \mathbf{I}_{\psi\psi}(\boldsymbol{\delta},\boldsymbol{\delta}) & \mathbf{0}_{3\times n_f} \\ \hline * & * & \mathbf{0}_{n_f\times n_f} \end{array}\right] \tag{5.18c}$$

It can be noticed that flexible dynamics augment the size of the previous rigid mass matrix in (4.8), and that the constant mass matrix encompasses the rigid motion in the first sub-blocks: $\mathbf{M}_\mathrm{f}(1{:}6, 1{:}6)$. A common way of writing a flexible mass matrix is to partition it according to the rigid and flexible dynamics, as follows:

$$\mathbf{M}_\mathrm{f} = \left[\begin{array}{c|c} \mathbf{M}_\mathrm{rr} & \mathbf{M}_\mathrm{rf} \\ \hline \mathbf{M}_\mathrm{fr} & \mathbf{M}_\mathrm{ff} \end{array}\right] \tag{5.19}$$

where sub-block $\mathbf{M}_\mathrm{rr}$ is actually the rigid mass matrix corrected by the flexible deformations, $\mathbf{M}_\mathrm{fr}$ is often referred to as the "modal participation" and denoted $\mathbf{L}$ (Alazard et al., 2008), and $\mathbf{M}_\mathrm{ff}$ gathers the "modal masses".

For the corrected rigid mass matrix , the equality $\mathbf{M}_\mathrm{rr} = \mathbf{M}_\mathrm{r}$ holds only for undeformed segments, i.e., with $\boldsymbol{\delta} = \mathbf{0}$, or equivalently $\mathbf{M}_\mathrm{r} = \mathbf{M}_\mathrm{f}^{(0)}(1{:}6, 1{:}6)$. Regarding the flexible modes, the second shape normalization proposed in (B.8) can simplify their dynamics by providing a diagonal stiffness matrix with the square of the pulsations. If the modes were completely decoupled (e.g., in the absence of payload with the clamped-free beam modes), the translational and rotary inertias $\bar{\mathbf{Z}}_{\boldsymbol{uu}} + \bar{\mathbf{Z}}_{\boldsymbol{\psi\psi}}$ would be diagonal. Nevertheless, they are full matrices in the general case of a loaded segment, meaning that the mode dynamics are coupled.

5.1.4 Potential Energy

Focusing on space applications with lightweight manipulators, gravity is neglected in the sequel. The segment potential energy is only due to its flexibility, which generates strain energy in each flexible direction. According to (De Luca and Siciliano, 1991; Mohan and Saha, 2009), the linear potential energy due to strain with the Euler-Bernoulli hypothesis reads[2]:

$$U = \frac{1}{2}\int_{\mathcal{B}} \left(ES \left(\frac{\partial u_x}{\partial s}\right)^2 + EI_z \left(\frac{\partial \psi_z}{\partial s}\right)^2 + EI_y \left(\frac{\partial \psi_y}{\partial s}\right)^2 + GI_p \left(\frac{\partial \psi_x}{\partial s}\right)^2 \right) \mathrm{d}s$$

where S denotes the section area, E the Young modulus, G the Poisson modulus, I_y and I_z the second moments of area, and I_p the polar second moment of area. Using (5.7a) and (5.7b), the following stiffness matrix $\mathbf{K}_{\delta\delta}$ appears:

$$\mathbf{K}_{\delta\delta} = \int_{\mathcal{B}} \begin{bmatrix} ES\boldsymbol{\phi}_x^{'}\boldsymbol{\phi}_x^{'\top} & \mathbf{0}_{n_x\times n_y} & \mathbf{0}_{n_x\times n_z} & \mathbf{0}_{n_x\times n_\alpha} \\ * & EI_z\,\boldsymbol{\phi}_y^{''}\boldsymbol{\phi}_y^{''\top} & \mathbf{0}_{n_y\times n_z} & \mathbf{0}_{n_y\times n_\alpha} \\ * & * & EI_y\,\boldsymbol{\phi}_z^{''}\boldsymbol{\phi}_z^{''\top} & \mathbf{0}_{n_z\times n_\alpha} \\ * & * & * & GI_p\,\boldsymbol{\phi}_\alpha^{'}\boldsymbol{\phi}_\alpha^{'\top} \end{bmatrix} \mathrm{d}s$$

in order to express the potential energy U as follows:

$$U = \frac{1}{2}\,\mathbf{x}_{\mathrm{o}}^{\top} \underbrace{\left[\begin{array}{cc|c} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n_f} \\ * & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n_f} \\ \hline * & * & \mathbf{K}_{\delta\delta} \end{array}\right]}_{\triangleq \mathbf{K}_{\mathrm{f}}} \mathbf{x}_{\mathrm{o}} \tag{5.20}$$

5.1.5 Dynamics

The quasi-Lagrangian equations are used to describe the dynamics of a single segment, based on their extension to flexible bodies by L. Meirovitch in (Meirovitch, 1991). As explained for rigid segments, the rigid motion is described by quasi-Lagrangian equations to simplify the rotational dynamics, while the flexible ones are derived by classic Lagrangian equations.

[2]More involved models could be considered with nonlinear stiffness matrices, as shown in (Mayo et al., 1995).

The complete set of equations applied to the Lagrangian $L = \quad T - U$ is given by:

$$\boldsymbol{Q}_{v_\mathrm{o}} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \boldsymbol{v}_\mathrm{o}}\right) - \left(\frac{\partial L}{\partial \mathbf{r}_\mathrm{o}}\right) \tag{5.21a}$$

$$\boldsymbol{Q}_{\omega_\mathrm{o}} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \boldsymbol{\omega}_\mathrm{o}}\right) - R_{\boldsymbol{\Psi}_\mathrm{o}}^{-\top}\left(\frac{\partial L}{\partial \boldsymbol{\Psi}_\mathrm{o}}\right) + \boldsymbol{v}_\mathrm{o}^{\times}\left(\frac{\partial L}{\partial \boldsymbol{v}_\mathrm{o}}\right) \tag{5.21b}$$

$$\boldsymbol{Q}_{\boldsymbol{\delta}} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\boldsymbol{\delta}}}\right) - \left(\frac{\partial L}{\partial \boldsymbol{\delta}}\right) \tag{5.21c}$$

where $\boldsymbol{Q}_{\boldsymbol{\delta}}$ are the generalized efforts corresponding to the flexible coordinates. Since the Lagrangian is still independent from the base position $\mathbf{r}_\mathrm{o}$ or attitude $\boldsymbol{\Psi}_\mathrm{o}$, their derivative vanish and the set of equations can be summarized by:

$$\boldsymbol{Q} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \mathbf{t}_\mathrm{o}}\right) - \frac{\partial \tilde{L}}{\partial \mathbf{x}_\mathrm{o}} \tag{5.22}$$

The computation of the state derivatives is performed for the case of the flexible segment constrained at both ends in Appendix B.3. Its expression is summarized as follows:

$$\frac{\partial \tilde{L}}{\partial \mathbf{x}_\mathrm{o}} = \underbrace{-\begin{bmatrix} \mathbf{0}_{3\times 1} \\ \boldsymbol{v}_\mathrm{o}^{\times}\left(\frac{\partial L}{\partial \boldsymbol{v}_\mathrm{o}}\right) \\ -\frac{\partial}{\partial \boldsymbol{\delta}}\left(T\right) \end{bmatrix}}_{\triangleq \tilde{\mathbf{M}}_\mathrm{f}\,\mathbf{t}_\mathrm{o}} \underbrace{-\begin{bmatrix} \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 1} \\ \frac{\partial}{\partial \boldsymbol{\delta}}\left(U\right) \end{bmatrix}}_{=\mathbf{K}_\mathrm{f}\mathbf{x}_\mathrm{o}} \tag{5.23}$$

where algebraic manipulations allows to express these derivatives as a matrix form by introducing $\tilde{\mathbf{M}}_\mathrm{f}$ as:

$$\tilde{\mathbf{M}}_\mathrm{f} = \left[\begin{array}{ccc|c} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & & \mathbf{0}_{3\times n_f} \\ \tilde{\mathbf{M}}_{\omega\mathrm{v}}(\boldsymbol{\omega}_\mathrm{o}, \boldsymbol{\delta}, \dot{\boldsymbol{\delta}}) & \mathbf{0}_{3\times 3} & & \mathbf{0}_{3\times n_f} \\ \hline \tilde{\mathbf{M}}_{\delta\mathrm{v}}(\boldsymbol{\omega}_\mathrm{o}) & \tilde{\mathbf{M}}_{\delta\omega}(\boldsymbol{\omega}_\mathrm{o}, \boldsymbol{\delta}) & & \tilde{\mathbf{M}}_{\delta\delta}(\boldsymbol{\omega}_\mathrm{o}) \end{array}\right] \tag{5.24}$$

It is shown that two of its sub-matrices yield a result similar to the rigid case, that allows to simplify the complete flexible dynamics:

$$\tilde{\mathbf{M}}_{\omega\mathrm{v}} = -\left(\mathring{\mathbf{M}}_{\omega v} + \boldsymbol{\omega}_\mathrm{o}^{\times}\mathbf{M}_{\omega v} - \mathbf{M}_{\omega v}\boldsymbol{\omega}_\mathrm{o}^{\times}\right) \tag{5.25a}$$

$$\tilde{\mathbf{M}}_{\delta\mathrm{v}} = -\left(-\mathbf{M}_{\delta v}\boldsymbol{\omega}_\mathrm{o}^{\times}\right) \tag{5.25b}$$

Deriving the flexible dynamics by exploiting the mass matrix decomposition in (5.18), the

same developments made for rigid bodies occur and the state equations are finally given by:

$$\begin{aligned}\boldsymbol{Q} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{M}_\mathrm{f}\,\mathbf{t}_\mathrm{o}\right) - \frac{\partial \tilde{L}}{\partial \mathbf{x}_\mathrm{o}} \\ &= \mathbf{M}_\mathrm{f}\,\dot{\mathbf{t}}_\mathrm{o} + \dot{\mathbf{M}}_\mathrm{f}\,\mathbf{t}_\mathrm{o} + \tilde{\mathbf{M}}_\mathrm{f}\,\mathbf{t}_\mathrm{o} + \mathbf{K}_\mathrm{f}\,\mathbf{x}_\mathrm{o}\end{aligned}$$

that is further simplified thanks to (5.25):

$$\boldsymbol{Q} = \mathbf{M}_\mathrm{f}\,\dot{\mathbf{t}}_\mathrm{o} + \left(\mathring{\mathbf{M}}_\mathrm{f} + \boldsymbol{\Omega}_\mathrm{o}\,\mathbf{M}_\mathrm{f} + \tilde{\mathbf{M}}_\mathrm{f}\right)\mathbf{E}_v\,\mathbf{t}_\mathrm{o} + \mathbf{K}_\mathrm{f}\,\mathbf{x}_\mathrm{o} \tag{5.26}$$

where $\boldsymbol{\Omega}_\mathrm{o}$ and $\mathbf{E}_v$ are augmented versions such that $\boldsymbol{\Omega}_\mathrm{o} = \mathrm{diag}(\boldsymbol{\omega}_\mathrm{o}^\times, \boldsymbol{\omega}_\mathrm{o}^\times, \mathbf{0}_{n_f \times n_f})$, and $\mathbf{E}_v = \mathrm{diag}(\mathbf{0}_{3\times 3}, \mathbf{I}_3, \mathbf{I}_{n_f})$. The time derivative of the mass matrix is defined in details in Appendix B.3, and can be summarized by:

$$\mathring{\mathbf{M}}_\mathrm{f} = \mathbf{M}_\mathrm{f}^{(1)}(\dot{\boldsymbol{\delta}}) + \mathbf{M}_\mathrm{f}^{(2)}(\dot{\boldsymbol{\delta}}, \boldsymbol{\delta}) + \mathbf{M}_\mathrm{f}^{(2)}(\boldsymbol{\delta}, \dot{\boldsymbol{\delta}}) \tag{5.27}$$

On the LHS, the generalized efforts yield almost the same expression as for the rigid case in (4.19). They are written in terms of speed as follows:

$$\boldsymbol{Q}_{q_j} = \left(\frac{\partial}{\partial \dot{q}_j}\begin{bmatrix}\boldsymbol{v}_\mathrm{o}^\top & \boldsymbol{\omega}_\mathrm{o}^\top\end{bmatrix}\right)\boldsymbol{w}_\mathrm{o} + \left(\frac{\partial}{\partial \dot{q}_j}\begin{bmatrix}\boldsymbol{v}_\mathrm{e}^\top & \boldsymbol{\omega}_\mathrm{e}^\top\end{bmatrix}\right)\boldsymbol{w}_\mathrm{e}$$

But the flexible twist is augmented in (5.9), such that the generalized effort increases in size when the flexible speeds are accounted for in $\dot{q}_j$. Using the expression of $\boldsymbol{v}_\mathrm{e}$ and $\boldsymbol{\omega}_\mathrm{e}$ obtained in (5.12), they read as follows:

$$\boldsymbol{Q} = \left(\frac{\partial}{\partial \mathbf{t}_\mathrm{o}}\begin{bmatrix}\boldsymbol{v}_\mathrm{o}^\top & \boldsymbol{\omega}_\mathrm{o}^\top\end{bmatrix}\right)\boldsymbol{w}_\mathrm{o} + \left(\frac{\partial}{\partial \mathbf{t}_\mathrm{o}}\begin{bmatrix}\boldsymbol{v}_\mathrm{e}^\top & \boldsymbol{\omega}_\mathrm{e}^\top\end{bmatrix}\right)\boldsymbol{w}_\mathrm{e} = \left[\frac{\mathbf{I}_6}{\mathbf{0}_{n_f\times 6}}\right]\boldsymbol{w}_\mathrm{o} + \mathbf{A}_{\mathrm{e,o}}^\top \boldsymbol{w}_\mathrm{e}$$

As proposed by A. Mohan and S. K. Saha in (Mohan and Saha, 2009), the global wrench at the segment base can be augmented by fictitious efforts:

$$\boldsymbol{w}_\mathrm{o} = \begin{bmatrix}\boldsymbol{f}_\mathrm{o}^\top & \boldsymbol{n}_\mathrm{o}^\top & \mathbf{0}_{n_f\times 1}^\top\end{bmatrix}^\top \tag{5.28}$$

in order to summarize the generalized efforts in a more compact way:

$$\boldsymbol{Q} = \boldsymbol{w}_\mathrm{o} + \mathbf{A}_{\mathrm{e,o}}^\top \boldsymbol{w}_\mathrm{e} \tag{5.29}$$

Flexible dynamics of a single segment

The dynamic equations of a flexible body are obtained for the reference frame $\mathcal{R}_\mathrm{o}$. This latter is considered as a floating frame rigidly fixed at the segment base to describe the flexible motion. Combining (5.26) and (5.29), it yields the following matrix form:

$$\mathbf{M}_\mathrm{f}\,\dot{\mathbf{t}}_\mathrm{o} + \left(\mathring{\mathbf{M}}_\mathrm{f} + \mathbf{\Omega}_\mathrm{o}\mathbf{M}_\mathrm{f} + \tilde{\mathbf{M}}_\mathrm{f}\right)\,\mathbf{E}_v\,\mathbf{t}_\mathrm{o} + \mathbf{K}_\mathrm{f}\,\mathbf{x}_\mathrm{o} = \boldsymbol{w}_\mathrm{o} + \mathbf{A}_{\mathrm{e,o}}^{\top}\,\boldsymbol{w}_\mathrm{e} \tag{5.30}$$

An advantage of this formulation is the reduction of numerical computations obtained by means of the matrix $\mathbf{E}_v$. Indeed, it avoids the first columns of $\mathbf{M}_\mathrm{f}(:,1{:}3)$ and of $\tilde{\mathbf{M}}_\mathrm{f}(:,1{:}3)$ to be evaluated. It is worth reminding the following matrix dependencies when the model is used in simulation, because they need to be updated at each time step:

$$\mathbf{M}_\mathrm{f}(\boldsymbol{\delta}), \quad \mathring{\mathbf{M}}_\mathrm{f}(\boldsymbol{\delta},\dot{\boldsymbol{\delta}}), \quad \tilde{\mathbf{M}}_\mathrm{f}(\boldsymbol{\omega}_\mathrm{o},\boldsymbol{\delta},\dot{\boldsymbol{\delta}}), \text{ and } \mathbf{A}_{\mathrm{e,o}}(\boldsymbol{\delta}).$$

Regarding the flexible dynamics, the model couples all the flexible modes, as mentioned above. This is in sharp contrast with the decoupled case of a clamped-free beam, where the flexible coordinates obey the classic dynamics of a second order oscillator with:

$$\ddot{\boldsymbol{\delta}} + 2\,[\xi]\,[\omega]\,\dot{\boldsymbol{\delta}} + \left[\omega^2\right]\,\boldsymbol{\delta} = -\mathbf{L}\begin{bmatrix}\boldsymbol{v}_\mathrm{o}\\ \boldsymbol{\omega}_\mathrm{o}\end{bmatrix}$$

where $[\xi]$ and $[\omega]$ denotes, respectively, the damping and pulsation of the flexible modes stored in a diagonal matrix, and $\mathbf{L} = \mathbf{M}_\mathrm{fr}$ are the modal participation. The nonlinear model developed here differs in twofold: first, the modal masses multiplying $\ddot{\boldsymbol{\delta}}$ are represented by full matrices with the translational and rotary inertias coupling the modes, and second, the nonlinear terms are included and may allow to experience the so-called "centrifugal stiffening" effect (Damaren and Sharf, 1995). This latter would appear if the arm is moving quickly, which is unlikely for space missions, or if a heavy payload is exciting the flexible modes, which is however far more realistic.

5.2 Model of a Flexible Multi-body System

In the same way as for the rigid case, the flexible dynamics of a whole manipulator is obtained by using recursively the previous model of a single segment. The same steps are described because flexibility is impacting the kinematics by additional translation and rotations, and kinetics and dynamics as well, since they derive from it. The advantage of the DeNOC models and algorithms derived in Chapter 4 is their direct extension to the flexible case by

only updating the key matrices and vectors:

$$\begin{aligned}
\mathbf{M}_{\mathrm{r}_i} &\Longrightarrow \mathbf{M}_{\mathrm{f}_i}(\boldsymbol{\delta}) \\
\mathbf{A}^{\mathrm{r}}_{i,i-1} &\Longrightarrow \mathbf{A}^{\mathrm{f}}_{i,i-1}(\boldsymbol{\delta}) \\
\mathbf{P}^{\mathrm{r}}_{i} &\Longrightarrow \mathbf{P}^{\mathrm{f}}_{i} \\
\boldsymbol{\gamma}_{\mathrm{r}_i} &\Longrightarrow \boldsymbol{\gamma}_{\mathrm{f}_i}(\boldsymbol{\delta}, \dot{\boldsymbol{\delta}})
\end{aligned}$$

To alleviate the notations, the flexible deflections expressed at the end-tip are denoted by $\mathbf{u}_i \triangleq \mathbf{u}_{\mathrm{e}_i}$ and $\boldsymbol{\psi}_i \triangleq \boldsymbol{\psi}_{\mathrm{e}_i}$, and their related matrices by $\boldsymbol{\Phi}_i \triangleq \boldsymbol{\Phi}_{\mathrm{e}_i}$ and $\boldsymbol{\Delta}_i \triangleq \boldsymbol{\Delta}_{\mathrm{e}_i}$.

In the sequel, the Newton-Euler scheme is adapted for the dynamics of flexible manipulators, but at the expense of a much higher computation time. Indeed, the previous matrices depend upon the flexible coordinates of their corresponding segment, and need to be updated accordingly as they vary during the simulation. Therefore, three different approximate models are proposed to limit the computational burden, while maintaining a high level of accuracy. They are compared based on the energy drift occurring during the simulation of two examples: a planar robot with 2 DoFs, and a Canadarm-like manipulator with 6 DoFs.

5.2.1 Kinematics

As illustrated in Figure 5.3, the base of a given segment is translated and rotated by the deflections at the end-tip of its predecessor. Hence, the homogeneous transformations presented in (4.21) must be updated by this additional motion due to flexibility. In addition to the set of four DH parameters, the flexible coordinates are used to describe the translation and rotation at the end-tip with the vector $\mathbf{u}_{i-1}$ and the Euler angles $\boldsymbol{\psi}_{i-1}$. The flexible homogeneous transformation ${}^{i-1}T^{\mathrm{f}}_i$ between $(i-1)^{th}$ and i^{th} segments is updated as follows :

$${}^{i-1}T^{\mathrm{f}}_i = \mathrm{Trans}(d_{i-1}, Z)\,\mathrm{Trans}(a_{i-1}, X)\,\mathrm{Trans}(\mathbf{u}_{i-1})\,\mathrm{Rot}(\boldsymbol{\psi}_{i-1})\,\mathrm{Rot}(\alpha_i, X)\,\mathrm{Rot}(\theta_i, Z)$$

where the flexible transformations are given by:

$$\mathrm{Trans}(\mathbf{u}_{i-1}) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{u}_{i-1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{5.31}$$

$$\mathrm{Rot}(\boldsymbol{\psi}_{i-1}) = \begin{bmatrix} {}^{\mathrm{o}_{i-1}}R_{\mathrm{e}_{i-1}} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{5.32}$$

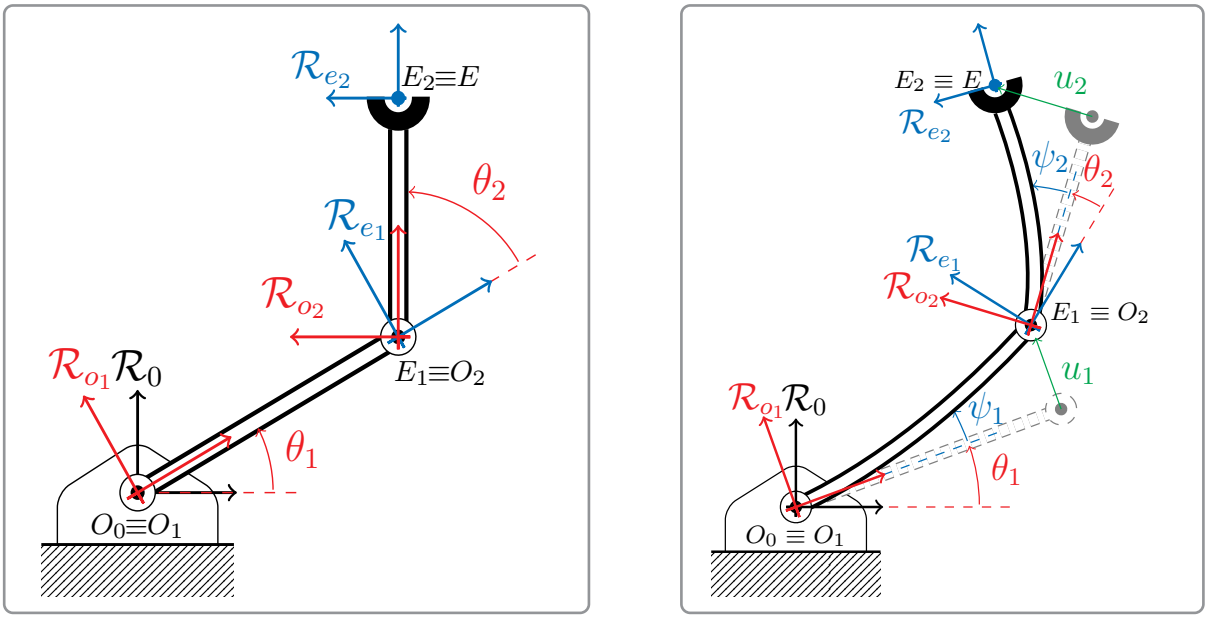


Figure 5.3: Robot kinematics comparison; (a) Rigid case, (b) Flexible case.

The end-tip deflection is expressed in $\mathcal{R}_{i-1}$ by $\mathbf{u}_{i-1} = \mathbf{\Phi}_{i-1}\,\boldsymbol{\delta}_{i-1}$ with (5.7a) and (5.12); and the end-tip rotation is described either by the Euler angles $\boldsymbol{\psi}_{i-1} = \mathbf{\Delta}_{i-1}\,\boldsymbol{\delta}_{i-1}$ with (5.7b), or by the corresponding rotation matrix ${}^{\mathrm{o}_{i-1}}R_{\mathrm{e}_{i-1}}$ given in (5.3).

The resulting rotation and translation between the successive frames $\mathcal{R}_{i-1} = \mathcal{R}_{\mathrm{o}_{i-1}}$ and $\mathcal{R}_i = \mathcal{R}_{\mathrm{o}_i}$ are given by ${}^{i-1}T_i^{\mathrm{f}}(1{:}3, 1{:}3)$ and ${}^{i-1}T_i^{\mathrm{f}}(1{:}3, 4)$. Using the rigid terms in (4.22) and (4.23), they read as follows :

$$ {}^{i-1}R_i^{\mathrm{f}} = {}^{\mathrm{o}_{i-1}}R_{\mathrm{e}_{i-1}}(\boldsymbol{\delta}_{i-1})\; {}^{i-1}R_i^{\mathrm{r}}(\theta_i) \tag{5.33} $$

$$ \mathbf{p}_{i-1}^{\mathrm{f}} = \mathbf{p}_{i-1}^{\mathrm{r}} + {}^{(i-1)}\mathbf{u}_{i-1}(\boldsymbol{\delta}_{i-1}) \tag{5.34} $$

where it is explicitly mentioned that the end-tip deflection $\mathbf{u}_{i-1}$ is expressed in $\mathcal{R}_{i-1}$, so the expression of $\mathbf{p}_{i-1}^{\mathrm{r}}$ must be taken in the same frame using (4.23).

The rotation matrix ${}^{i-1}R_i^{\mathrm{f}}$ can be decomposed like the mass matrix, i.e., according to its dependency upon $\boldsymbol{\delta}_{i-1}$. The nominal value corresponds to the undeformed state, with the rigid rotation ${}^{i-1}R_i^{\mathrm{r}}(\theta_i)$, while the flexible corrective term ${}^{\mathrm{o}_{i-1}}R_{\mathrm{e}_{i-1}}$ is function of $\boldsymbol{\psi}_{i-1}$, which depends upon $\boldsymbol{\delta}_{i-1}$ by (5.7b). Hence, the decomposition of the rotation matrix reads as follows:

$$ {}^{i-1}R_i^{\mathrm{f}} = {}^{i-1}R_i^{\mathrm{f}\,(0)} + {}^{i-1}R_i^{\mathrm{f}\,(1)}(\boldsymbol{\delta}_{i-1}) $$

with

$$ {}^{i-1}R_i^{\mathrm{f}\,(0)} = {}^{i-1}R_i^{\mathrm{r}} \tag{5.35a} $$

$$ {}^{i-1}R_i^{\mathrm{f}\,(1)}(\boldsymbol{\delta}_{i-1}) = \left({}^{\mathrm{o}_{i-1}}R_{\mathrm{e}_{i-1}}(\boldsymbol{\delta}_{i-1}) - \mathbf{I}_3\right) {}^{i-1}R_i^{\mathrm{r}} \tag{5.35b} $$

Considering that the flexible coordinates are additional degrees of freedom, the kinematics depends upon generalized coordinates which are no longer scalar terms but vectors:

$$ \boldsymbol{q}_i = \begin{bmatrix} \theta_i \\ \boldsymbol{\delta}_i \end{bmatrix} \in \mathbb{R}^{1+n_{f_i}} \tag{5.36} $$

such that the global vector of generalized coordinates is now given by:

$$ \mathbf{q}_{\mathrm{f}} = \begin{bmatrix} \boldsymbol{q}_1 & \dots & \boldsymbol{q}_n \end{bmatrix}^\top \tag{5.37} $$

Therefore, the i^{th} flexible transformation ${}^{i-1}T_i^{\mathrm{f}}$ has been written as a function of $(\theta_i, \boldsymbol{\delta}_{i-1})$. Assuming that the base is rigid (i.e., $\boldsymbol{\delta}_0 = \emptyset$), the inertial position of any point M along the i^{th} link is obtained with the same recursive relation as (4.25):

$$ \begin{aligned} \tilde{\mathbf{r}}_M &= \left({}^0T_1^{\mathrm{f}}(\theta_1)\, {}^1T_2^{\mathrm{f}}(\theta_2, \boldsymbol{\delta}_1)\, \dots {}^{i-1}T_i^{\mathrm{f}}(\theta_i, \boldsymbol{\delta}_{i-1})\right) {}^i\tilde{\mathbf{p}}_M^{\mathrm{f}}(\boldsymbol{\delta}_i) \\ &= {}^0T_i^{\mathrm{f}}(\mathbf{q}_{\mathrm{f}})\, {}^i\tilde{\mathbf{p}}_M^{\mathrm{f}}(\boldsymbol{\delta}_i) \end{aligned} \tag{5.38} $$

where the relative position ${}^i\mathbf{p}_M^{\mathrm{f}}$ must account for the flexible deflection of the i^{th} segment using (5.4) and (5.7):

$$ {}^i\mathbf{p}_M^{\mathrm{f}}(\boldsymbol{\delta}_i) = {}^i\mathbf{p}_M + {}^i\mathbf{u}_M(\boldsymbol{\delta}_i) $$

To describe the end-effector kinematics, the extra transformation ${}^nT_{\mathrm{E}}$ now depends upon $\boldsymbol{\delta}_n$, as shown in Figure 5.3b with the frame $\mathcal{R}_{\mathrm{e}_2}$ w.r.t. $\mathcal{R}_{\mathrm{o}_2}$. The global homogeneous transformation is then given by a function of $\mathbf{q}_{\mathrm{f}}$ by:

$$ {}^0T_E^{\mathrm{f}}(\mathbf{q}_{\mathrm{f}}) = {}^0T_1^{\mathrm{f}}(\theta_1)\, {}^1T_2^{\mathrm{f}}(\theta_2, \boldsymbol{\delta}_1)\, \dots {}^{n-1}T_n^{\mathrm{f}}(\theta_n, \boldsymbol{\delta}_{n-1})\, {}^nT_{\mathrm{E}}(\boldsymbol{\delta}_n) $$

Thanks to these notations, the effector frame is described by the same equation as (4.26):

$$ \mathbf{r}_{\mathrm{E}}(\mathbf{q}_{\mathrm{f}}) = {}^0T_E^{\mathrm{f}} \begin{bmatrix} \mathbf{0}_{3\times 1} \\ 1 \end{bmatrix} \qquad \text{and} \qquad {}^0R_{\mathrm{E}}^{\mathrm{f}}(\mathbf{q}_{\mathrm{f}}) = {}^0T_{\mathrm{E}}^{\mathrm{f}}(1{:}3, 1{:}3) \tag{5.39} $$

which is equivalent to the general kinematic relation:

$$\mathbf{x}_{\mathrm{E}} = \begin{bmatrix} \mathbf{r}_{\mathrm{E}} \\ \boldsymbol{\Psi}_{\mathrm{E}} \end{bmatrix} = f_{kin}(\mathbf{q}_{\mathrm{f}}) \tag{5.40}$$

5.2.2 Kinetics

Still using the twist propagation relation at the scale of one single segment in (5.12), the recursive twist relation for a rigid manipulator in (4.28) is extended to flexible ones. One major change is the increase in the number of DoF, as $\boldsymbol{q}_i$ is a vector instead of a scalar. Using the twist-propagation matrix for the end-tip of a flexible segment in (5.12), the linear and angular velocities of the i^{th} segment are still obtained by:

$$\begin{bmatrix} \boldsymbol{v}_{\mathrm{o}_i} \\ \boldsymbol{\omega}_{\mathrm{o}_i} \end{bmatrix} = \mathbf{t}_{\mathrm{e}_{i-1}} + \dot{\theta}_i \, \mathbf{z}_i = \mathbf{A}_{\mathrm{e}_{i-1},\mathrm{o}_{i-1}} \, \mathbf{t}_{\mathrm{o}_{i-1}} + \dot{\theta}_i \, \mathbf{z}_i$$

Then, according to the flexible twist definition in (5.9) and to the expression of the segment generalized coordinates in (5.36), an extra relation must be added in the form of: $\dot{\boldsymbol{\delta}}_i = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\delta}_i$, to eventually obtain:

$$\underbrace{\begin{bmatrix} \boldsymbol{v}_{\mathrm{o}_i} \\ \boldsymbol{\omega}_{\mathrm{o}_i} \\ \dot{\boldsymbol{\delta}}_i \end{bmatrix}}_{=\mathbf{t}_i} = \underbrace{\begin{bmatrix} \mathbf{A}_{\mathrm{e}_{i-1},\mathrm{o}_{i-1}} \\ \mathbf{0}_{n_{f_i}\times(6+n_{f_{i-1}})} \end{bmatrix}}_{=\mathbf{A}_{i,i-1}} \underbrace{\mathbf{t}_{\mathrm{o}_{i-1}}}_{=\mathbf{t}_{i-1}} + \underbrace{\begin{bmatrix} \mathbf{z}_i & \mathbf{0}_{6\times n_{f_i}} \\ \mathbf{0}_{n_{f_i}\times 1} & \mathbf{I}_{n_{f_i}} \end{bmatrix}}_{=\mathbf{P}_i} \underbrace{\begin{bmatrix} \dot{\theta}_i \\ \dot{\boldsymbol{\delta}}_i \end{bmatrix}}_{=\dot{\boldsymbol{q}}_i} \tag{5.41}$$

The same relation as (4.28) is then obtained for flexible systems, provided that the twist-propagation matrices $\mathbf{A}_{i,i-1}$ and the joint rate propagation matrices $\mathbf{P}_i$ are defined as follows:

$$\mathbf{A}^{\mathrm{f}}_{i,i-1} = \left[\begin{array}{ccc} \mathbf{I}_3 & -\left(\mathbf{p}^{\mathrm{f}}_{i-1}(\boldsymbol{\delta}_{i-1})\right)^{\times} & \boldsymbol{\Phi}_{i-1} \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 & R_{\boldsymbol{\psi}_{i-1}} \boldsymbol{\Delta}_{i-1} \\ \hline \mathbf{0}_{n_{f_i}\times 3} & \mathbf{0}_{n_{f_i}\times 3} & \mathbf{0}_{n_{f_i}\times n_{f_{i-1}}} \end{array}\right] \tag{5.42a}$$

$$\mathbf{P}^{\mathrm{f}}_{i} = \left[\begin{array}{c|c} \mathbf{z}_i & \mathbf{0}_{6\times n_{f_i}} \\ \hline \mathbf{0}_{n_{f_i}\times 1} & \mathbf{I}_{n_{f_i}} \end{array}\right] \tag{5.42b}$$

The same partitioning used for the mass matrix in (5.17) applies for the twist-propagation matrix $\mathbf{A}^{\mathrm{f}}_{i,i-1}$, since it explicitly depends upon the flexible coordinates. On the other hand, the joint rate propagation matrix $\mathbf{P}_i$ remains constant in $\mathcal{R}_i$ and does not present flexible corrective terms. The decomposition of $\mathbf{A}^{\mathrm{r}}_{i,i-1}$ is straightforward since the segment vector $\mathbf{p}^{\mathrm{f}}_{i-1}$ is already written as the summation of a nominal (rigid) value and of a corrective term in (5.34). The twist-propagation matrix is then split using (5.7a) to explicit the dependency upon $\boldsymbol{\delta}_{i-1}$:

$$\mathbf{A}^{\mathrm{f}}_{i,i-1} = \mathbf{A}^{\mathrm{f}\,(0)}_{i,i-1} + \mathbf{A}^{\mathrm{f}\,(1)}_{i,i-1}(\boldsymbol{\delta}_{i-1}) \tag{5.43}$$

with

$$\mathbf{A}^{\mathrm{f}\,(0)}_{i,i-1} = \left[\begin{array}{ccc} \mathbf{I}_3 & -\left(\mathbf{p}^{\mathrm{r}}_{i-1}\right)^{\times} & \boldsymbol{\Phi}_{i-1} \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \boldsymbol{\Delta}_{i-1} \\ \hline \mathbf{0}_{n_{f_i}\times 3} & \mathbf{0}_{n_{f_i}\times 3} & \mathbf{0}_{n_{f_i}\times n_{f_{i-1}}} \end{array}\right] \tag{5.44a}$$

$$\mathbf{A}^{\mathrm{f}\,(1)}_{i,i-1}(\boldsymbol{\delta}_{i-1}) = \left[\begin{array}{ccc} \mathbf{0}_{3\times 3} & -\left(\boldsymbol{\Phi}_{i-1}\boldsymbol{\delta}_{i-1}\right)^{\times} & \mathbf{0}_{3\times n_{f_{i-1}}} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \left(R_{\boldsymbol{\psi}_{i-1}} - \mathbf{I}_3\right)\boldsymbol{\Delta}_{i-1} \\ \hline \mathbf{0}_{n_{f_i}\times 3} & \mathbf{0}_{n_{f_i}\times 3} & \mathbf{0}_{n_{f_i}\times n_{f_{i-1}}} \end{array}\right] \tag{5.44b}$$

In the same way, the generalized rotation matrix from $\mathcal{R}_{i-1}$ to $\mathcal{R}_i$ is re-defined to encompass the flexible coordinates, and is also split according to corrective terms. In order to use the same expressions in

$$\mathbf{A}_{i,i-1} = {}^{i-1}\boldsymbol{R}_i^{\top}\ {}^{(i-1)}\mathbf{A}^{\mathrm{f}}_{i,i-1} \tag{5.45a}$$

$$\mathbf{P}_i = {}^{(i)}\mathbf{P}^{\mathrm{f}}_i \tag{5.45b}$$

with the extended rotation matrix:

$${}^{i-1}\boldsymbol{R}_i = \mathrm{diag}\left({}^{i-1}R^{\mathrm{f}}_i,\ {}^{i-1}R^{\mathrm{f}}_i,\ \mathbf{I}_{n_{f_i}}\right) \tag{5.46}$$

As mentioned above, extending the same matrices from the rigid case to the flexible one allows to define kinetics and dynamics with less efforts. As an example, the definition of the Jacobian matrix in (4.33) still holds for flexible manipulators, but includes now the flexible coordinates:

$$\mathbf{t}_{\mathrm{E}} = \left[\begin{array}{c} \boldsymbol{v}_{\mathrm{E}} \\ \boldsymbol{\omega}_{\mathrm{E}} \end{array}\right] = \mathbf{J}_{\mathrm{E}}(\mathbf{q}_{\mathrm{f}})\,\dot{\mathbf{q}}_{\mathrm{f}} \tag{5.47}$$

5.2.3 Constrained Dynamics of the Segments

The time derivative of the twist provided in (4.36) is still valid, but the time derivative of the twist-propagation matrix must be re-developed. Indeed, it is no longer constant in $\mathcal{R}_{i-1}$, but rather depends explicitly upon time through $\boldsymbol{\delta}_{i-1}$ in (5.44b). Using the vectrix time derivative in (4.14b), one obtains:

$$\dot{\mathbf{A}}^{\mathrm{f}}_{i,i-1} = \mathring{\mathbf{A}}^{\mathrm{f}}_{i,i-1} + \begin{bmatrix} \boldsymbol{\Omega}_{i-1}\,\mathbf{A}^{\mathrm{f}}_{i,i-1}(1{:}6,:) \\ \hline \mathbf{0}_{n_{f_i}\times(6+n_{f_{i-1}})} \end{bmatrix} - \left[\begin{array}{c|c} \mathbf{A}^{\mathrm{f}}_{i,i-1}(:,1{:}6)\,\boldsymbol{\Omega}_{i-1} & \mathbf{0}_{(6+n_{f_i})\times n_{f_{i-1}}} \end{array}\right]$$

$$\dot{\mathbf{P}}^{\mathrm{f}}_{i} = \begin{bmatrix} \boldsymbol{\Omega}_i\,\mathbf{P}^{\mathrm{f}}_{i}(1{:}6,:) \\ \mathbf{0}_{n_{f_i}\times(1+n_{f_i})} \end{bmatrix}$$

where $\boldsymbol{\Omega}_i$ is augmented by a zero matrix in (5.25). Depending whether the product is done on the right/left hand side, matrices $\boldsymbol{\Omega}_i = \mathrm{diag}(\boldsymbol{\omega}_i^{\times}, \boldsymbol{\omega}_i^{\times}, \mathbf{0}_{n_{f_i}})$ and $\bar{\boldsymbol{\Omega}}_i = \mathrm{diag}(\boldsymbol{\omega}_i^{\times}, \boldsymbol{\omega}_i^{\times}, \mathbf{0}_{n_{f_{i-1}}})$ are introduced. The time derivatives are re-written as follows:

$$\dot{\mathbf{A}}^{\mathrm{f}}_{i,i-1} = \mathring{\mathbf{A}}^{\mathrm{f}}_{i,i-1} + \boldsymbol{\Omega}_{i-1}\,\mathbf{A}^{\mathrm{f}}_{i,i-1} - \mathbf{A}^{\mathrm{f}}_{i,i-1}\bar{\boldsymbol{\Omega}}_{i-1} \tag{5.48a}$$

$$\dot{\mathbf{P}}^{\mathrm{f}}_{i} = \boldsymbol{\Omega}_i\,\mathbf{P}^{\mathrm{f}}_{i} \tag{5.48b}$$

With these notations, the decomposition of $\dot{\mathbf{A}}^{\mathrm{f}}_{i,i-1}$ into its nominal and corrective terms yields:

$$\dot{\mathbf{A}}^{\mathrm{f}}_{i,i-1} = \dot{\mathbf{A}}^{\mathrm{f}\,(0)}_{i,i-1} + \dot{\mathbf{A}}^{\mathrm{f}\,(1)}_{i,i-1}(\boldsymbol{\delta}_{i-1}) \tag{5.49}$$

where the nominal term is given explicitly by:

$$\dot{\mathbf{A}}^{\mathrm{f}\,(0)}_{i,i-1} = \boldsymbol{\Omega}_{i-1}\,\mathbf{A}^{\mathrm{f}\,(0)}_{i,i-1} - \mathbf{A}^{\mathrm{f}\,(0)}_{i,i-1}\,\bar{\boldsymbol{\Omega}}_{i-1} \tag{5.50a}$$

$$= \left[\begin{array}{cc|c} \mathbf{0}_{3\times3} & \left(\boldsymbol{\omega}^{\times}_{i-1}\mathbf{p}^{\mathrm{r}}_{i-1}\right)^{\times} & \boldsymbol{\omega}^{\times}_{i-1}\boldsymbol{\Phi}_{i-1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{\omega}^{\times}_{i-1}\boldsymbol{\Delta}_{i-1} \\ \hline \mathbf{0}_{n_{f_i}\times3} & \mathbf{0}_{n_{f_i}\times3} & \mathbf{0}_{n_{f_i}\times n_{f_{i-1}}} \end{array}\right]$$

while the corrective one reads:

$$\dot{\mathbf{A}}^{\mathrm{f}\,(1)}_{i,i-1}(\boldsymbol{\delta}_{i-1}) = \mathring{\mathbf{A}}^{\mathrm{f}\,(1)}_{i,i-1} + \boldsymbol{\Omega}_{i-1}\,\mathbf{A}^{\mathrm{f}\,(1)}_{i,i-1} - \mathbf{A}^{\mathrm{f}\,(1)}_{i,i-1}\,\bar{\boldsymbol{\Omega}}_{i-1} \tag{5.50b}$$

$$= \left[\begin{array}{cc|c} \mathbf{0}_{3\times3} & -\left(\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{\Phi}_{i-1}\boldsymbol{\delta}_{i-1}\right)\right)^{\times} & \mathbf{0}_{3\times n_{f_{i-1}}} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{\mathrm{d}}{\mathrm{d}t}\left(\left(R_{\boldsymbol{\psi}_{i-1}} - \mathbf{I}_3\right)\boldsymbol{\Delta}_{i-1}\right) \\ \hline \mathbf{0}_{n_{f_i}\times3} & \mathbf{0}_{n_{f_i}\times3} & \mathbf{0}_{n_{f_i}\times n_{f_{i-1}}} \end{array}\right]$$

with

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{\Phi}_{i-1}\boldsymbol{\delta}_{i-1}\right) = \boldsymbol{\Phi}_{i-1}\dot{\boldsymbol{\delta}}_{i-1} + \boldsymbol{\omega}_{i-1}\times\boldsymbol{\Phi}_{i-1}\boldsymbol{\delta}_{i-1}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\left(R_{\boldsymbol{\psi}_{i-1}} - \mathbf{I}_3\right)\boldsymbol{\Delta}_{i-1}\right) = \left(\dot{R}_{\boldsymbol{\psi}_{i-1}} + \boldsymbol{\omega}^{\times}_{i-1}\left(R_{\boldsymbol{\psi}_{i-1}} - \mathbf{I}_3\right)\right)\boldsymbol{\Delta}_{i-1}$$

where the expression of $\dot{R}_{\boldsymbol{\psi}_{i-1}}$ is given by (A.12) in Appendix A.2.

The final expressions of the time derivative of twist-propagation and joint-rate-propagation matrices are obtained by applying the change of frame presented in (5.46), as done for the rigid case in (4.38).

$$\dot{\mathbf{A}}_{i,i-1} = {}^{i-1}\boldsymbol{R}_i^{\top}\,{}^{(i-1)}\dot{\mathbf{A}}^{\mathrm{f}}_{i,i-1} \tag{5.51a}$$

$$\dot{\mathbf{P}}_i = \boldsymbol{\Omega}_i\,{}^{(i)}\mathbf{P}^{\mathrm{f}}_i \tag{5.51b}$$

Gathering all the previous results, the constrained dynamics of a flexible segment is defined by bringing together the free dynamics equation in (5.30), the recursive twist computation and its derivative in (4.28) and (4.36), updated with the flexible twist-propagation and joint-rate-propagation matrices in (5.45) and (5.51). Reminding that $\boldsymbol{w}_{\mathrm{e}_i} = -\boldsymbol{w}_{\mathrm{o}_{i+1}}$, and denoting $\boldsymbol{w}_i \triangleq \boldsymbol{w}_{\mathrm{o}_i}$, the fundamental equation of motion of a constrained flexible segment is given by:

$$\boxed{\mathbf{M}_{\mathrm{f}_i}\,\dot{\mathbf{t}}_i + \boldsymbol{\gamma}_{\mathrm{f}_i} = \boldsymbol{w}_i - \mathbf{A}^{\top}_{i+1,i}\,\boldsymbol{w}_{i+1}} \tag{5.52}$$

with the local vector of Coriolis and centrifugal forces reading:

$$\boldsymbol{\gamma}_{\mathrm{f}_i} = \left(\mathring{\mathbf{M}}_{\mathrm{f}_i} + \boldsymbol{\Omega}_i\mathbf{M}_{\mathrm{f}_i} + \tilde{\mathbf{M}}_{\mathrm{f}_i}\right)\mathbf{E}_v\,\mathbf{t}_i + \mathbf{K}_{\mathrm{f}_i}\,\mathbf{x}_i \tag{5.53}$$

The final step of inverse dynamics is the computation of the generalized efforts. Once again, their derivation is straightforward thanks to the DeNOC framework, where flexible kinetics and dynamics are based on the same matrices as the rigid case. Hence, the computations introduced in (4.42) are still valid with the updated twist-propagation and joint-

rate-propagation matrix in (5.42a) and (5.42b), and using the extended wrench in (5.28). Nevertheless, since the flexible generalized coordinates $\boldsymbol{q}_i$ lie in $\mathbb{R}^{(1+n_f)}$, their corresponding efforts are vectors of the same size:

$$\boldsymbol{\tau}_i = \mathbf{P}_i^\top \boldsymbol{w}_i \tag{5.54}$$

The flexible dynamics of each segment has been derived through the update of the twist recursion with (5.42) and (5.48), and on wrenches and generalized efforts in (5.52) and (5.54). Therefore, the inverse dynamics scheme and the algorithms presented earlier remain the same for the flexible manipulator, provided that all matrices and their time derivatives are updated.

5.2.4 Inverse Dynamics

In this section, the inverse dynamics algorithm based on the Newton-Euler scheme is updated, and the closed-form model is derived for the flexible case. These steps were thoroughly covered for rigid manipulators, and they easily extend to flexible dynamics through the DeNOC approach. A main contribution is provided though with the derivation of approximate models for simulation. Indeed, if hypothesis are made on the expression of key matrices, as $\mathbf{M}_{\mathrm{f}}$, $\mathbf{A}^{\mathrm{f}}_{i,i-1}$, and ${}^{i-1}\boldsymbol{R}^{\mathrm{f}}_i$, their effect is coherently propagated through all dynamic steps. Three approximate models are presented, and the best compromise is chosen between the computation time and the resulting accuracy.

Newton-Euler recursive scheme The double recursion is still used to derive the inverse dynamics with the recursive Newton-Euler scheme. The algorithm presented in Algorithm 5.1 with the function `InvDynFlex` is an updated version of `InvDynRig` in Algorithm 4.1 with the new equation labels of the flexible section. In addition, the algorithms lines involving the decomposition of matrices $\mathbf{M}_{\mathrm{f}}$, $\mathbf{A}^{\mathrm{f}}_{i,i-1}$, and ${}^{i-1}\boldsymbol{R}^{\mathrm{f}}_i$, are marked with "$\star$" and will be used to derive the approximate models.

Firstly, the outward kinetic loop still starts with: $\mathbf{t}_0 = \mathbf{0}_{6\times 1}$ and $\dot{\mathbf{t}}_0 = \mathbf{0}_{6\times 1}$, but the matrices $\mathbf{A}^{\mathrm{f}}_{1,0}$ and $\dot{\mathbf{A}}^{\mathrm{f}}_{1,0}$ must be augmented at the bottom by $\mathbf{0}_{n_{f_1}\times 6}$, in order to include the flexible states of the first segment. The following notation is used to alleviate the algorithm description: $\mathbf{I}_{m_{f_1}\times 6} = \begin{bmatrix} \mathbf{I}_6^\top & \mathbf{0}_{n_{f_1}\times 6}^\top \end{bmatrix}^\top$, where $m_{f_i} \triangleq 6 + n_{f_i}$ is the size of the mass matrix $\mathbf{M}_{\mathrm{f}_i}$, or the LHS size of the twist-propagation $\mathbf{A}_{i,i-1}$.

Secondly, the inward dynamic loop is initialized with the external efforts applied by the payload at the end-effector. The model used for the payload may be changed to include a flexible behavior. Reminding that the mass matrix of the payload is denoted by $\mathbf{M}_{n+1} = \mathbf{M}_{payload}$, and its twist by $\mathbf{t}_{n+1} = \mathbf{t}_{payload}$, the last wrench $\boldsymbol{w}_{n+1}$ is initialized by:

$$\boldsymbol{w}_{n+1} = \mathbf{M}_{n+1}\,\dot{\mathbf{t}}_{n+1} + \boldsymbol{\gamma}_{n+1} \tag{5.55}$$

where $\boldsymbol{\gamma}_{n+1}$ is defined for a rigid payload by (4.48), and for a flexible model by the general relation in (5.53):

$$\boldsymbol{\gamma}^{\mathrm{r}}_{n+1} = \boldsymbol{\Omega}_{n+1}\,\mathbf{M}_{\mathrm{r}_{n+1}}\,\mathbf{E}_v\,\mathbf{t}_{n+1} \tag{5.56}$$

or (5.57)

$$\boldsymbol{\gamma}^{\mathrm{f}}_{n+1} = \left(\mathring{\mathbf{M}}_{\mathrm{f}_{n+1}} + \boldsymbol{\Omega}_{n+1}\,\mathbf{M}_{\mathrm{f}_{n+1}} + \tilde{\mathbf{M}}_{\mathrm{f}_{n+1}}\right)\mathbf{E}_v\,\mathbf{t}_{n+1} + \mathbf{K}_{\mathrm{f}_{n+1}}\,\mathbf{x}_{n+1} \tag{5.58}$$

The function `InvDynFlex` is still used to compute the Coriolis and centrifugal vector $\mathbf{h}_{\mathrm{f}}$ and the stiffness terms in a numerically efficient way by:

$$\texttt{InvDynFlex}(\mathbf{q}_{\mathrm{f}}, \dot{\mathbf{q}}_{\mathrm{f}}, \mathbf{0}) = \left(\mathbf{h}_{\mathrm{f}}(\mathbf{q}_{\mathrm{f}}, \dot{\mathbf{q}}_{\mathrm{f}}) + \mathbf{K}\,\mathbf{q}_{\mathrm{f}}\right) \tag{5.59}$$

These terms are detailed in the next section with the closed-loop equation of flexible dynamics.

Closed-form dynamics Computations performed for the rigid case are straightforward to extend to the flexible case thanks to the DeNOC approach. In this respect, the Coriolis and centrifugal terms are denoted by the following abuse of notation, already used in (A.25) of Appendix A.3.3:

$$\dot{\mathbf{M}}_{\mathrm{f}_i} \triangleq \left(\mathring{\mathbf{M}}_{\mathrm{f}_i} + \boldsymbol{\Omega}\,\mathbf{M}_{\mathrm{f}_i} + \tilde{\mathbf{M}}_{\mathrm{f}_i}\right)\mathbf{E}_v \tag{5.60}$$

The system of constrained equations of motion in (5.52) is then written in the similar matrix form, using the same block diagonal matrices and two additional ones for flexible Coriolis terms and stiffness ones :

$$[\dot{\mathbf{M}}_{\mathrm{f}}] = \mathrm{diag}(\dot{\mathbf{M}}_{\mathrm{f}_i},\, i = 1 \ldots n)$$
$$[\mathbf{K}_{\mathrm{f}}] = \mathrm{diag}(\mathbf{K}_{\mathrm{f}_i},\, i = 1 \ldots n)$$

The manipulator dynamics is then described similarly to the rigid case in (5.61). Merging all the free coordinates into $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 & \ldots & \mathbf{x}_n \end{bmatrix}^\top$, the manipulator dynamics is described by

Algorithm 5.1: Newton-Euler algorithm for flexible inverse dynamics (`InvDynFlex`)

Function : $\boldsymbol{\tau}_\mathrm{f} = \texttt{InvDynFlex}(\mathbf{q}_\mathrm{f},\, \dot{\mathbf{q}}_\mathrm{f},\, \ddot{\mathbf{q}}_\mathrm{f})$
Input : $\mathbf{q}_\mathrm{f},\, \dot{\mathbf{q}}_\mathrm{f},\, \ddot{\mathbf{q}}_\mathrm{f}$
Output : $\boldsymbol{\tau}_\mathrm{f}$
Data : $\left\{ {}^{i-1}R_i\,,\ \mathbf{A}^{\mathrm{f}\,(0)}_{i,i-1}\,,\ \mathbf{A}^{\mathrm{f}\,(1)}_{i,i-1}\,,\ \mathbf{P}_i\,,\ \mathbf{M}_{\mathrm{f}_i}\ ;\ i = 1 \dots n+1 \right\}$

`Initialize` $\left(\mathbf{t}_0 = \dot{\mathbf{t}}_0 = \mathbf{0}_{6\times 1},\ \mathbf{A}^{\mathrm{f}}_{1,0} = \mathbf{I}_{m_{f_1}\times 6},\ \dot{\mathbf{A}}^{\mathrm{f}}_{1,0} = \mathbf{0}_{m_{f_1}\times 6}\right)$ // **KINETICS LOOP**

for $i = 1 \dots n+1$ **do** // Recursive computation of $\mathbf{t}_i$, $\dot{\mathbf{t}}_i$, and $\boldsymbol{\Omega}_i$

- ${}^{i}R_{i-1} = \left({}^{i-1}R^{\mathrm{f}\,(0)}_i(q_i) + {}^{i-1}R^{\mathrm{f}\,(1)}_i(\boldsymbol{\delta}_{i-1}) \right)^\top$ ⋆(5.35) [a]
- ${}^{i}\boldsymbol{R}_{i-1} = \mathrm{diag}({}^{i}R_{i-1}, {}^{i}R_{i-1})$ (5.46)
- $\mathbf{A}_{i,i-1} = {}^{i}\boldsymbol{R}_{i-1}\, \mathbf{A}^{\mathrm{f}}_{i,i-1}$ (5.45a)
- $\dot{\mathbf{A}}_{i,i-1} = {}^{i}\boldsymbol{R}_{i-1}\, \dot{\mathbf{A}}^{\mathrm{f}}_{i,i-1}$ (5.51a)
- $\mathbf{t}_i = \mathbf{A}_{i,i-1}\, \mathbf{t}_{i-1} + \mathbf{P}_i\, \dot{q}_i$ (5.41) [b]
- $\boldsymbol{\omega}_i = \mathbf{t}_i(4:6)$ (5.9)
- $\boldsymbol{\Omega}_i = \mathrm{diag}(\boldsymbol{\omega}_i^\times,\, \boldsymbol{\omega}_i^\times)$
- $\dot{\mathbf{t}}_i = \mathbf{A}_{i,i-1}\, \dot{\mathbf{t}}_{i-1} + \dot{\mathbf{A}}_{i,i-1}\, \mathbf{t}_{i-1} + \mathbf{P}_i\, \ddot{q}_i + \boldsymbol{\Omega}_i\, \mathbf{P}_i\, \dot{q}_i$ (4.36)
- **if** $i \leq n$ **then**
  - $\mathbf{A}^{\mathrm{f}}_{i+1,i} = \mathbf{A}^{\mathrm{f}\,(0)}_{i+1,i} + \mathbf{A}^{\mathrm{f}\,(1)}_{i+1,i}(\boldsymbol{\delta}_i)$ ⋆(5.43)
  - $\dot{\mathbf{A}}^{\mathrm{f}}_{i+1,i} = \mathring{\mathbf{A}}^{\mathrm{f}}_{i+1,i} + \boldsymbol{\Omega}_i\, \mathbf{A}^{\mathrm{f}}_{i+1,i} - \mathbf{A}^{\mathrm{f}}_{i+1,i}\, \bar{\boldsymbol{\Omega}}_i$ ⋆(5.49)
- **end**

end

`Initialize` $\left(\boldsymbol{w}_{n+1} = \mathbf{M}_{n+1}\, \dot{\mathbf{t}}_{n+1} + \boldsymbol{\gamma}_{n+1}\right)$ // **DYNAMICS LOOP**

for $i = n \dots 1$ **do** // Recursive computation of $\boldsymbol{w}_i$, τ_i

- $\mathbf{M}_{\mathrm{f}_i} = \mathbf{M}^{(0)}_{\mathrm{f}_i} + \mathbf{M}^{(1)}_{\mathrm{f}_i}(\boldsymbol{\delta}_i) + \mathbf{M}^{(2)}_{\mathrm{f}_i}(\boldsymbol{\delta}_i, \boldsymbol{\delta}_i)$ ⋆(5.18)
- $\mathring{\mathbf{M}}_{\mathrm{f}_i} = \mathbf{M}^{(1)}_{\mathrm{f}_i}(\dot{\boldsymbol{\delta}}_i) + \mathbf{M}^{(2)}_{\mathrm{f}_i}(\dot{\boldsymbol{\delta}}_i, \boldsymbol{\delta}_i) + \mathbf{M}^{(2)}_{\mathrm{f}_i}(\boldsymbol{\delta}_i, \dot{\boldsymbol{\delta}}_i)$ ⋆(5.27)
- $\tilde{\mathbf{M}}_{\mathrm{f}_i} = \tilde{\mathbf{M}}_{\mathrm{f}_i}(\boldsymbol{\delta}_i, \boldsymbol{\omega}_i)$ ⋆(5.24)
- $\boldsymbol{\gamma}_{\mathrm{f}_i} = \left(\mathring{\mathbf{M}}_{\mathrm{f}_i} + \boldsymbol{\Omega}_i\, \mathbf{M}_{\mathrm{f}_i} + \tilde{\mathbf{M}}_{\mathrm{f}_i} \right) \mathbf{E}_v\, \mathbf{t}_i + \mathbf{K}_{\mathrm{f}_i}\, \mathbf{x}_i$ ⋆(5.53)
- $\boldsymbol{W}_i = \mathbf{M}_{\mathrm{f}_i} \dot{\mathbf{t}}_i + \boldsymbol{\gamma}_{\mathrm{f}_i}$ (5.52)
- $\boldsymbol{w}_i = \boldsymbol{W}_i + \mathbf{A}^\top_{i+1,i} \boldsymbol{w}_{i+1}$
- $\boldsymbol{\tau}_i = \mathbf{P}_i^\top\, \boldsymbol{w}_i$ (5.54)

end

return $\boldsymbol{\tau}_\mathrm{f}$

[a] For the payload rotation at $i = n+1$, the nominal rotation matrix is supposed constant, and does not depend upon the fictitious variable q_{n+1}. For the first rotation at $i = 1$, the base is supposed rigid, so $\boldsymbol{\delta}_0$ does not exist, and the corrective rotation matrix is ${}^{0}R^{\mathrm{f}\,(1)}_1 = \mathbf{0}_{3\times 3}$.

[b] For the payload at $i = n+1$, the fictitious variables $\dot{q}_{n+1} = \ddot{q}_{n+1} = 0$ are introduced, with $\mathbf{P}_{n+1} = \mathbf{0}_{6\times 1}$.

the following system of $(6n + \sum_i n_{f_i})$ equations:

$$[\mathbf{M}_\mathrm{f}]\ \dot{\mathbf{t}} + [\dot{\mathbf{M}}_\mathrm{f}]\,\mathbf{t} + [\mathbf{K}_\mathrm{f}]\ \mathbf{x} = \mathbf{N}_l^{-\top}\,\boldsymbol{w} + \overline{\mathbf{A}}_{n+1,n}^{\top}\,\boldsymbol{w}_\mathrm{E} \tag{5.61}$$

The motion constraints are now applied by pre-multiplying by $\mathbf{N}^\top$ to obtain the reduced equations of motion in the Kane's formalism. As it was done in the rigid case, the twist and its time derivative are developed under the DeNOC form with matrix $\mathbf{N}$. The new system of equations is then reduced to describe the dynamics of the $(n + \sum_i n_{f_i})$ generalized coordinates, as follows:

$$\mathbf{N}^\top\,[\mathbf{M}_\mathrm{f}]\left(\mathbf{N}\ddot{\mathbf{q}}_\mathrm{f} + \dot{\mathbf{N}}\dot{\mathbf{q}}_\mathrm{f}\right) + \mathbf{N}^\top[\dot{\mathbf{M}}_\mathrm{f}]\mathbf{N}\,\dot{\mathbf{q}}_\mathrm{f} + \mathbf{N}^\top\ [\mathbf{K}_\mathrm{f}]\ \mathbf{x} = \mathbf{N}^\top\mathbf{N}_l^{-\top}\,\boldsymbol{w} + \mathbf{N}^\top\overline{\mathbf{A}}_{n+1,n}^{\top}\,\boldsymbol{w}_\mathrm{E}$$

A similar closed-form appears for the flexible case, since kinetics and dynamics were written using the same notations of twist-propagation and joint-rate-propagation matrices in (5.41), and of mass matrix and nonlinear terms in (5.52). The principle that internal efforts do not work still applies and simplifies the RHS to obtain the generalized efforts obtained in (5.54), and developed more thoroughly in (A.18a) (Mohan and Saha, 2009). The Jacobian matrix is also multiplying the effector wrench, but under its flexible form given in (5.47).

A last computation must be performed to obtain the stiffness matrix. As seen in (5.20), the segment stiffness matrix $\mathbf{K}_{\mathrm{f}_i}$ is artificially augmented to be expressed as a function of the free coordinates $\mathbf{x}_i$. Indeed, the term $[\mathbf{K}_\mathrm{f}]\ \mathbf{x}$ could be expressed as a column vector merging all the segment stiffness forces given by: $\left[\ \mathbf{0}_{6\times1}^\top \quad (\mathbf{K}_{\delta_i\delta_i}\boldsymbol{\delta}_i)^\top\ \right]^\top$. Hence, this vector remains the same when multiplied by $\mathbf{N}_l^\top$, because of the structure of $\mathbf{A}_{i,j}$ matrices in (5.42a). Finally, the last multiplication by $\mathbf{N}_d^\top$ only reduces the size of the zero vector part from 6 to 1, i.e., the dimension of the rigid generalized coordinates of one segment. Finally, denoting by $\mathbf{K}_i$ the segment stiffness matrices, the global one is obtained by merging them diagonally as:

$$\mathbf{K} = \mathrm{diag}\,(\mathbf{K}_i\,,\ i = 1\ldots n) \tag{5.62a}$$

$$\mathbf{K}_i = \begin{bmatrix} 0 & \mathbf{0}_{1\times n_{f_i}} \\ \mathbf{0}_{n_{f_i}\times1} & \mathbf{K}_{\delta_i\delta_i} \end{bmatrix} \tag{5.62b}$$

The flexible dynamic model is now derived based on these last results. It is important to recall that the algorithms provided in Appendix A.3 to compute the mass matrix and the Coriolis and centrifugal vector are still valid for the flexible case.

Flexible dynamics of a robotic manipulator

The set of free-body equations in (5.61) yields the manipulator dynamics under the classic closed form :

$$\mathbf{D}_{\mathrm{f}}(\mathbf{q}_{\mathrm{f}})\,\ddot{\mathbf{q}}_{\mathrm{f}} + \mathbf{C}_{\mathrm{f}}(\mathbf{q}_{\mathrm{f}},\dot{\mathbf{q}}_{\mathrm{f}})\,\dot{\mathbf{q}}_{\mathrm{f}} + \mathbf{K}\,\mathbf{q}_{\mathrm{f}} = \boldsymbol{\tau}_{\mathrm{f}} + \mathbf{J}_{\mathrm{E}}^{\top}\,\boldsymbol{w}_{\mathrm{E}} \tag{5.63}$$

with

$$\mathbf{D}_{\mathrm{f}}(\mathbf{q}_{\mathrm{f}}) = \mathbf{N}^{\top}\,[\mathbf{M}_{\mathrm{f}}]\,\mathbf{N} \tag{5.64}$$

$$\mathbf{C}_{\mathrm{f}}(\mathbf{q}_{\mathrm{f}},\dot{\mathbf{q}}_{\mathrm{f}}) = \mathbf{N}^{\top}\,[\mathbf{M}_{\mathrm{f}}]\,\dot{\mathbf{N}} + \mathbf{N}^{\top}\left([\mathring{\mathbf{M}}_{\mathrm{f}}] + [\boldsymbol{\Omega}]\,[\mathbf{M}_{\mathrm{f}}] + [\tilde{\mathbf{M}}_{\mathrm{f}}]\right)[\mathbf{E}_{v}]\,\mathbf{N} \tag{5.65}$$

$$\boldsymbol{\tau}_{\mathrm{f}} = \mathbf{N}_{d}^{\top}\,\boldsymbol{w} \tag{5.66}$$

5.2.5 Forward Dynamics

In the flexible case, the forward dynamics algorithm remains exactly the same thanks to the DeNOC approach. Only the size of the involved vectors and matrices will change in the algorithm presented in Appendix A.3.4.

The forward dynamics scheme is illustrated in Figure 5.4. Firstly, inverse dynamics is computed by Algorithm 5.1 with $\ddot{\mathbf{q}}_{\mathrm{f}} = \mathbf{0}$ to obtain $\mathbf{h}_{\mathrm{f}} + \mathbf{K}\mathbf{q}_{\mathrm{f}}$. During these computations, matrices $\mathbf{A}_{i,i-1}$, $\mathbf{P}_i$, and $\mathbf{M}_{\mathrm{f}_i}$ are stored after being updated with their respective corrective terms, given by (5.35b) and (5.44b) for $\mathbf{A}^{\mathrm{f}\,(1)}_{i,i-1}$ and ${}^{i-1}R^{\mathrm{f}\,(1)}_{i}$, and by (5.18) for $\mathbf{M}^{(1)}_{\mathrm{f}_i}$ and $\mathbf{M}^{(2)}_{\mathrm{f}_i}$. Their expression is then used in the forward dynamics algorithm given in Appendix A.3.4 based on (Saha, 1999; Mohan and Saha, 2009). It does not need to invert the whole mass matrix, and only inverts a reduced inertia of size $(1+n_{f_i})$ at each step of the RGE. As shown in (Mohan and Saha, 2007, 2009), this approach proves to be numerically more efficient **and** more stable than most of the available algorithms in the literature.

In the next section, the numerical scheme is simplified by neglecting some of these corrective terms in order to reduce the on-line computations during simulation. Doing this, much attention is paid to the resulting accuracy of the approximate simulations.

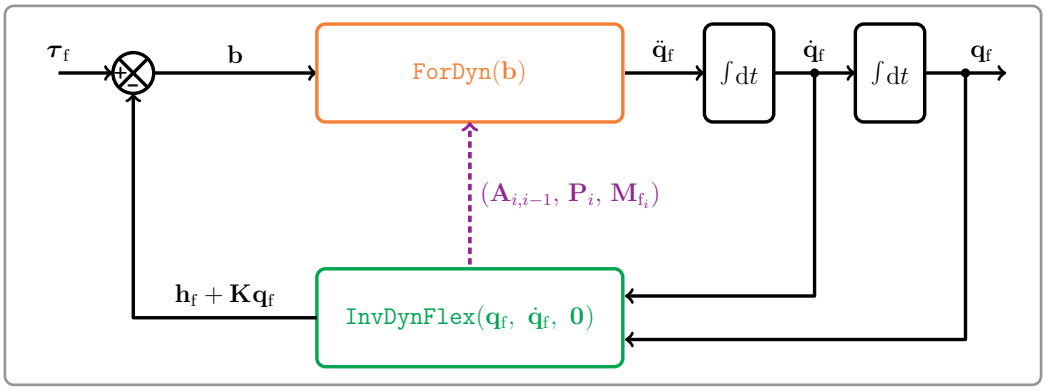


Figure 5.4: Forward dynamics scheme for numerical simulation of flexible multi-body systems

5.2.6 Approximate Dynamics for Simulation

Simulation of flexible manipulators increase significantly the computational burden compared to rigid systems. The main reason is the update of the mass matrix at each time step, and the computation of highly nonlinear terms in the Coriolis and centrifugal vector. By carefully neglecting the least important corrective terms, approximate models can be used to reduce the computation time. This is in sharp contrast with (Mohan and Saha, 2009), where only integral expressions are provided and the influence of these corrections is not explicit. Moreover, even in (Sharf and Damaren, 1992; Damaren and Sharf, 1995), similar approximations are made but are uncorrelated between the mass matrix and the nonlinear forcing terms. Thanks to (5.63), the inertial approximations are propagated more coherently on both the mass matrix and the forcing terms. In order to reduce these costly computations, three different approximate models are built based on some hypothesis at the kinematic, kinetic and dynamic levels (Sharf and Damaren, 1992; Damaren and Sharf, 1995).

The main idea is to neglect the terms of first and second order w.r.t. the flexible coordinates, and to observe the resulting effects in simulation. This analysis depends strongly on the trajectory speed and on the presence of a payload, since both will excite the flexible modes and thus increase the influence of their corrective terms on dynamics. For each approximate model, one of these terms is neglected: either the first order one in the expression of ${}^{i-1}R_i$ in (5.35) and $\mathbf{A}_{i,i-1}$ in (5.44), or the first and second order ones for $\mathbf{M}_\mathrm{f}$ in (5.18a). Four different models are investigated in simulation: a reference case includes all the corrective terms on $\mathbf{M}_{\mathrm{f}_i}$, $\mathbf{A}_{i,i-1}$ and ${}^{i-1}R_i$, while three approximate models neglect some kinematic and dynamic terms to check their respective influence, as follows:

- `flex` : reference case, all corrective terms are computed;
- `flex-M` : segment mass matrices ($\mathbf{M}_{\mathrm{f}_i}$) are constant w.r.t. $\boldsymbol{\delta}$;
- `flex-MA` : mass matrices and twist-propagation matrices ($\mathbf{M}_{\mathrm{f}_i}, \mathbf{A}_{i,i-1}$) are constant w.r.t. $\boldsymbol{\delta}$;
- `flex-MR` : mass matrices and rotation matrices ($\mathbf{M}_{\mathrm{f}_i}, {}^{i-1}R_i$) are constant w.r.t. $\boldsymbol{\delta}$.

Each of this model is quickly covered in the sequel to highlight the implications of these hypothesis on flexible dynamics. Using the thorough derivation of the flexible model in previous sections, the repercussions of these hypothesis are accounted for in a coherent way along all the modeling steps. Indeed, they affect either the kinetic energy or the kinematic constraints, and thus induce strong changes in the dynamic equation in (5.63).

5.2.6.1 Approximate models

As illustrated in Figure 5.4, the simulation scheme is based on the computation in series of inverse dynamics to obtain the nonlinear forcing terms in $\mathbf{h}_{\mathrm{f}}$, and then, of forward dynamics to inverse the mass matrix by the RGE technique. Doing so, the approximations made on the kinematic and dynamic matrices impact consistently the computation of the forcing terms as well as the inversion of the mass matrix. In the following, the hypothesis are made at the inverse dynamics level, keeping in mind that the resulting mass matrices and twist-propagation matrices are used then in forward dynamics.

Reference case `flex`
All corrective terms are computed, including the first and second order terms in the time derivatives of $\mathbf{M}_{\mathrm{f}_i}$ and $\mathbf{A}_{i,i-1}$ used to express $\mathbf{h}_{\mathrm{f}}$.

Constant mass matrices `flex-M`
The mass matrix is supposed to remain constant, assuming that $\mathbf{M}_{\mathrm{f}_i}^{(1)} + \mathbf{M}_{\mathrm{f}_i}^{(2)} \ll \mathbf{M}_{\mathrm{f}_i}^{(0)}$ over time. This hypothesis involves that $\mathring{\mathbf{M}}_{\mathrm{f}_i} = \mathbf{0}$ vanishes, and that $\tilde{\mathbf{M}}_{\mathrm{f}_i}$ reduces to its $\tilde{\mathbf{M}}_{\omega\mathrm{v}}$ term in (5.24). Indeed, the differentiation of the mass matrix w.r.t. the flexible coordinates produces the terms $\tilde{\mathbf{M}}_{\delta,\mathrm{v}/\omega\delta}$, so they cancel if $\mathbf{M}_{\mathrm{f}_i}$ is constant. Eventually, the nonlinear terms in $\boldsymbol{\gamma}_{\mathrm{f}_i}$ cannot be simplified with $\mathbf{E}_v$ and read:

$$\boldsymbol{\gamma}_i = \left(\boldsymbol{\Omega}_i \, \mathbf{M}_{\mathrm{f}_i} - \boldsymbol{\Omega}_i \, \mathbf{M}_{\mathrm{f}_i} + \tilde{\mathbf{M}}_{\mathrm{f}_i}\right) \mathbf{t}_i$$

By contrast with the other models, the kinematic corrective terms in ${}^{i-1}R_i$ and $\mathbf{A}_{i,i-1}$ are maintained.

As it will be seen from the simulation results, the mass matrix hypothesis yields a very accurate model, showing that the dynamic corrective terms are not the most important ones to consider. Therefore, the next two models are still using this hypothesis, with an additional one on the kinematic or kinetic corrective terms.

Constant twist-propagation matrices `flex-MA`

In addition to the constant mass matrix hypothesis, the twist-propagation matrix $\mathbf{A}_{i,i-1}$ is reduced to its nominal value in this model. It is assumed that $\mathbf{A}_{i,i-1}^{\mathrm{f}\,(1)} \ll \mathbf{A}_{i,i-1}^{\mathrm{f}\,(0)}$. This additional assumption implies that $\dot{\mathbf{A}}_{i,i-1}$ is derived similarly to the rigid case, by restricting it to $\dot{\mathbf{A}}_{i,i-1}^{\mathrm{f}\,(0)}$ in (5.50a). The kinematic corrective terms in ${}^{i-1}R_i$ are kept.

Constant rotation matrices `flex-MR`

In addition to the constant mass matrix hypothesis, the flexible rotations induced by bending and torsion are neglected in this last model. The rotation matrix is given by the product of this flexible rotation induced by the previous segment, with its nominal value ${}^{i-1}R_i^{\mathrm{f}\,(0)}$ in (5.33), i.e., the rigid-body rotation. Nevertheless, the kinetic corrective terms in $\mathbf{A}_{i,i-1}$ are kept for this case.

5.2.6.2 Simulation test cases

Two classic examples taken from the literature are used to validate the dynamic scheme, and to evaluate the accuracy of approximate models. The simulation results allow to quantify the effect of each corrective term and its impact on the computation time, as well as the error introduced by the approximations. The ultimate goal is to recommend the simulation model leading to the best compromise between accuracy and computation time.

Data used for both models are given in details through Appendix D.1. On the one hand, simulations are led for a planar robot with hubs and payload taken from (De Luca and Siciliano, 1991). Its physical data do not have any physical meaning, but it was used to validate the dynamic model obtained with the previous inverse and forward dynamics, with the symbolic one developed in (De Luca and Siciliano, 1991). On the other hand, a 6DoF manipulator moving in 3D is also used to study the influence of the corrective terms for a more complex case. Rigid data are taken from (Mohan and Saha, 2009), while the flexible data are computed using the AMM presented in Appendix B.1 since they were not provided.

Planar Manipulator A first set of simulations is performed for a planar robot with 2DoF, as illustrated in Figure 5.3. Only bending in the plane is considered for this example. As mentioned above, data from (De Luca and Siciliano, 1991) do not have a physical meaning and result in overestimated rotary inertias in $\bar{\mathbf{Z}}_{\psi\psi}$. Therefore, they must be neglected here to be compatible with the symbolic model provided in (De Luca and Siciliano, 1991), which does not account for these terms.

Two modes are considered on each segment, leading thus to four flexible variables on the whole manipulator. As explained in (De Luca and Siciliano, 1991), a nominal configuration must be chosen to obtain the resulting loads of each segment, and then to compute their clamped-loaded modes. The extended position is chosen here, with $\boldsymbol{\theta}_{nom} = \mathbf{0}_{2\times1}$. Using the physical data provided in Appendix D.1.1, the following pulsations are obtained in rad/s for the first two bending modes of each segment:

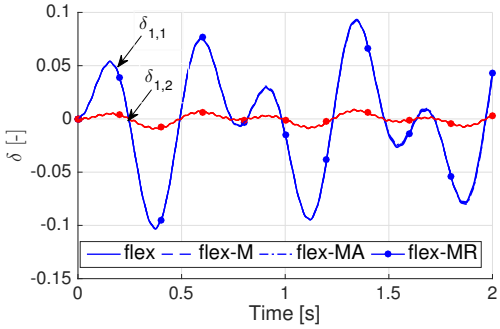
i	$\omega_{i,1}$	$\omega_{i,2}$
2	3.01	11.29
3	13.67	99.99

The following stiffness matrices are then obtained for each segment, reminding that mode shapes are normalized to ensure that $\mathbf{K}_{f_i}(j,j) = m_i\,\omega_{i,j}^2$ (De Luca and Siciliano, 1991):

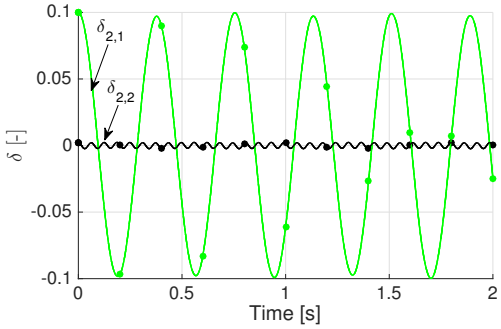
$$\mathbf{K}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.91 & 0 \\ 0 & 0 & 12.74 \end{bmatrix} \qquad \mathbf{K}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 18.73 & 0 \\ 0 & 0 & 999.88 \end{bmatrix}$$

These results were computed using Appendix B.1, and prove to be consistent with the data provided in (De Luca and Siciliano, 1991).

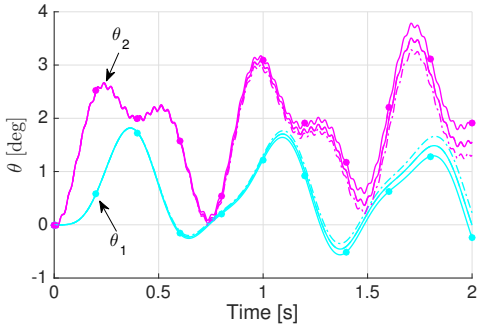
In simulation, the first test case from (De Luca and Siciliano, 1991) is used to test the free flexible dynamics: no torques are applied at both joints, and non-zero initial conditions are set on the flexible coordinates of the second segment $\boldsymbol{\delta}_2$. Initial conditions are given by: $\theta_1 = \theta_2 = \delta_{1,1} = \delta_{1,2} = 0$ and $\delta_{2,1} = 0.1$, $\delta_{2,2} = 0.002$. Due to the high coupling between both segments, and due to the absence of control, they are vibrating freely. The results are presented in Figure 5.5, and discussed in Section 5.2.6.3.



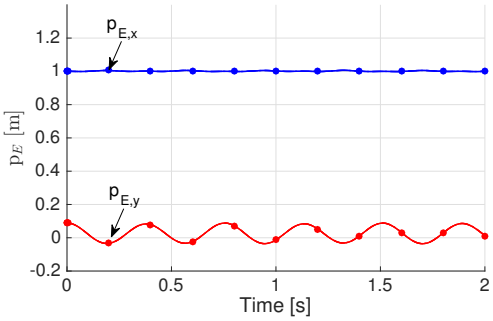
(a) Flexible coordinates $\boldsymbol{\delta}_1$



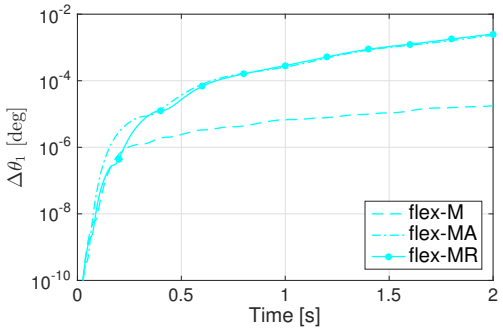
(b) Flexible coordinates $\boldsymbol{\delta}_2$



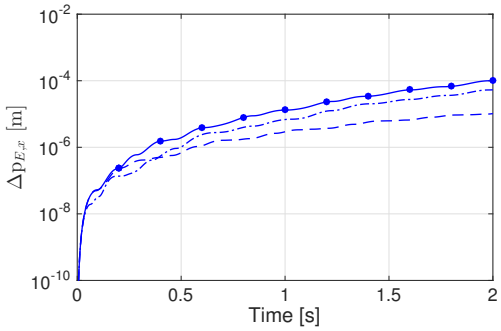
(c) Joint Angles $\boldsymbol{\theta}$



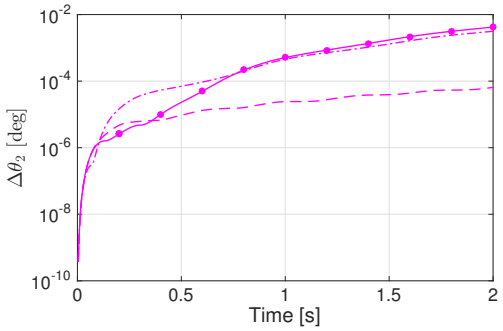
(d) Effector Position $\mathbf{p}_E$



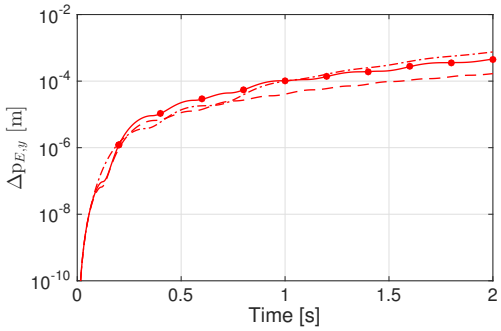
(e) Approximation Error for θ_1



(f) Approximation Error for $\mathbf{p}_{E,x}$



(g) Approximation Error for θ_2



(h) Approximation Error for $\mathbf{p}_{E,y}$

Figure 5.5: Simulation results for the planar example.

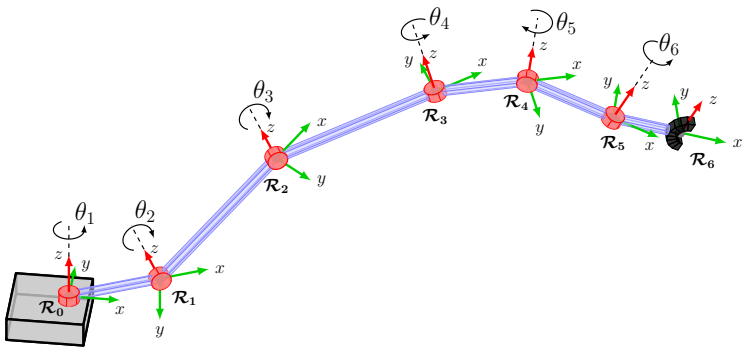


Figure 5.6: Classic Denavit-Hartenberg frames for the Canadarm manipulator of (Mohan and Saha, 2009) (with Z axis in red)

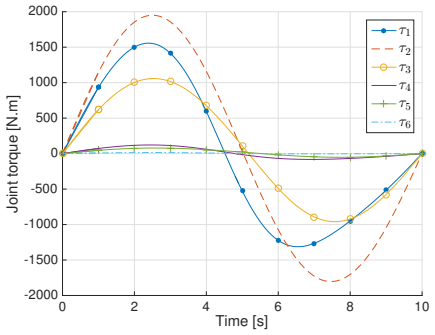


Figure 5.7: Joint torques for the Canadarm example

Canadarm Manipulator The *Space Shuttle Remote Manipulator System*, called Canadarm for brevity, is used to illustrate 3D motion and flexibility in each direction. Its kinematic scheme with the classical DH frames is given in Figure 5.6 and its data are summarized in Appendix D.1.2. Using the model given in (Mohan and Saha, 2009), only the second and third segments are considered flexible. Ten modes are used for both segments, one in traction, four in each bending direction and one in torsion. Since no flexible data were available in this reference, they were computed with the theory introduced in appendix B.1. The extended position is also chosen to compute the flexible modes, with $\boldsymbol{\theta}_{nom} = \mathbf{0}_{6\times1}$.

With these parameters, the following pulsations are obtained in rad/s for the flexible modes of the second and third segments:

i	$\omega_{i,x,1}$	$\omega_{i,y,1}$	$\omega_{i,y,2}$	$\omega_{i,y,3}$	$\omega_{i,y,4}$	$\omega_{i,\alpha,1}$
2	127.03	1.07	4.83	42.17	112.98	215.12
3	149.01	2.16	11.22	44.35	115.55	252.35

Simulations are performed for a forced dynamics scenario given in (Damaren and Sharf, 1995) and (Mohan and Saha, 2009). The following "pick-and-place" trajectories are desired over time:

$$\theta_i^d(t) = \theta_{des}\left(\frac{t}{T} - \frac{1}{2\pi}\sin\left(\frac{2\pi}{T}t\right)\right) \quad , \quad \forall i \in [1;\, 6]$$

with $\theta_{des} = 0.5\,rad/s$ the desired final joint angles, starting from 0, and T is the trajectory duration. Using the previous inverse dynamics model `InvDynFlex` in Algorithm 5.1, the torques required to follow this trajectory are computed and their profile is given in Figure 5.7.

Eventually, the simulation of the 6DoF Canadarm is presented in Figures 5.8 and 5.9 and discussed in the next section.

5.2.6.3 Simulation results

Accuracy of the approximate models is estimated by checking the global energy evolution along time, as proposed by (Damaren and Sharf, 1995). Theoretically, this energy is assumed to stay constant in the free dynamics case. On the contrary, the system power, defined as its time derivative, is supposed to match the external power applied by the joint torques in the forced dynamics case. The input power corresponding to the joint actuators and its related work are defined by:

$$\mathcal{P}_{input} = \sum_{i=1}^{6} \tau_i \, \dot{\theta}_i \qquad\qquad \mathcal{W}_{input} = \int_0^t \mathcal{P}_{input} \mathrm{d}t$$

Theoretically, this work should match exactly the relative system energy denoted by $\Delta\mathcal{E} = \mathcal{E}(t) - \mathcal{E}_0$. The global system energy is defined by:

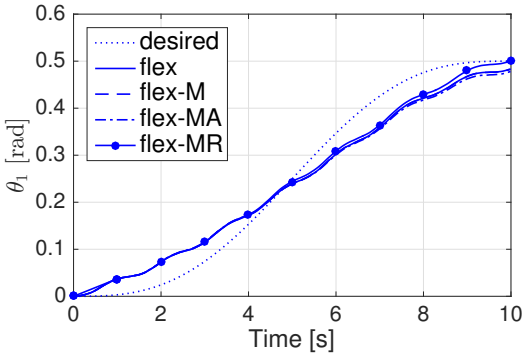
$$\mathcal{E} = \frac{1}{2} \, \dot{\mathbf{q}}^\top \mathbf{D}(\mathbf{q}) \, \dot{\mathbf{q}} + \frac{1}{2} \, \mathbf{q}^\top \mathbf{K} \, \mathbf{q}$$

To measure this energy drift, the *Root Mean Square (RMS)* indicator introduced in (Damaren and Sharf, 1995) is used. The peak energy $\mathcal{E}_{peak}$ reached during the simulation is introduced to weight this energy drift. The numerical error and its RMS value over the whole time horizon T are defined by:

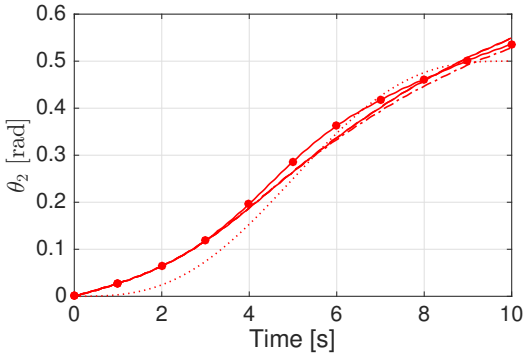
$$e_{\mathcal{E}}(t) = \frac{\Delta\mathcal{E} - \mathcal{W}_{input}}{|\mathcal{E}_{peak}|} \qquad\qquad e_{\mathcal{E},RMS} = \sqrt{\frac{\int_0^T \left(e_{\mathcal{E}}(t)\right)^2 \mathrm{d}t}{T}}$$

Coming back to the previous simulation results for both examples, conclusions are twofold:

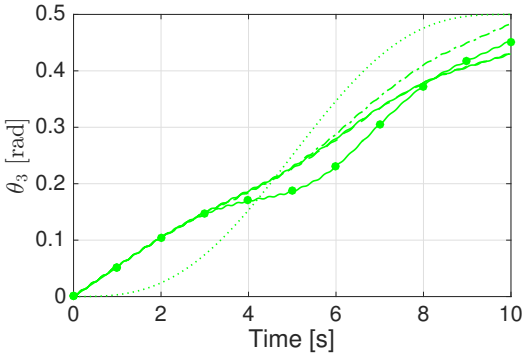
• **Validation of Flexible Dynamics**: the algorithms for inverse and forward dynamics are validated by the two examples taken from (De Luca and Siciliano, 1991) and (Mohan and Saha, 2009). In the planar case, curves are exactly matching the simulations provided by De Luca and Siciliano, which are based on symbolic dynamic equations. In addition, a symbolic computation of the mass matrix and of the Coriolis and centrifugal vector has been carried out with the algorithms provided in Appendix A.3 and it yields exactly the same terms with the first order truncation proposed in (De Luca and Siciliano, 1991). In the second example, the joint angles behaviors remain close but not exactly the same, because



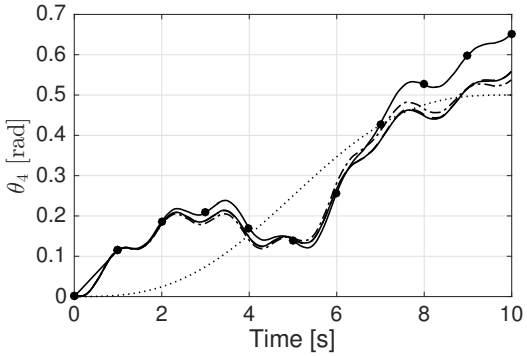
(a) Joint angle θ_1



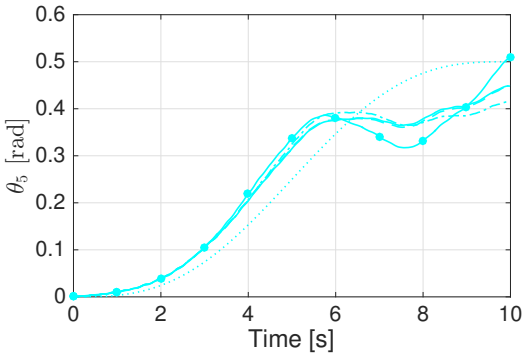
(b) Joint angle θ_2



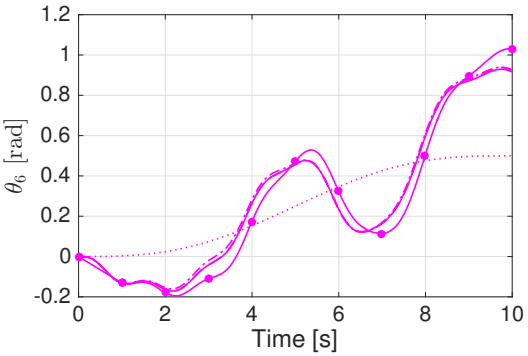
(c) Joint angle θ_3



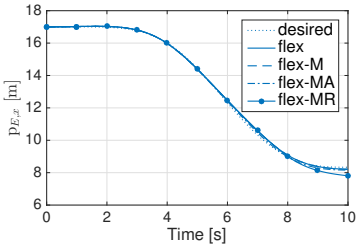
(d) Joint angle θ_4



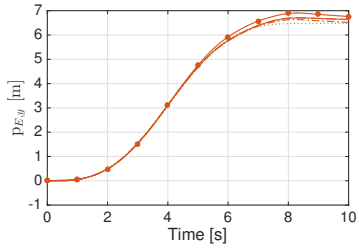
(e) Joint angle θ_5



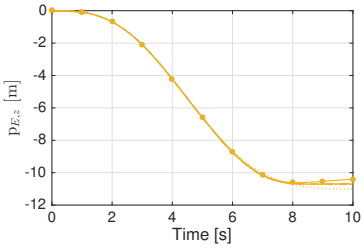
(f) Joint angle θ_6



(g) Effector Position $\mathbf{p}_{E,x}$

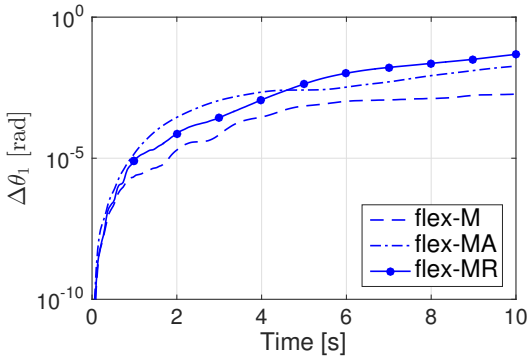


(h) Effector Position $\mathbf{p}_{E,y}$

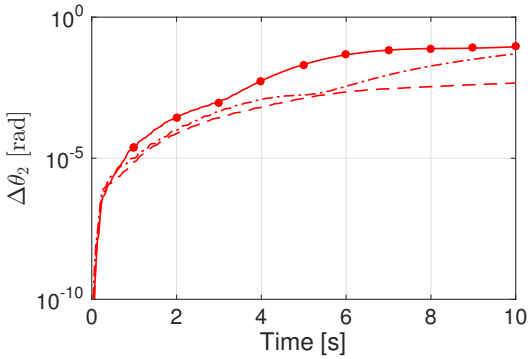


(i) Effector Position $\mathbf{p}_{E,z}$

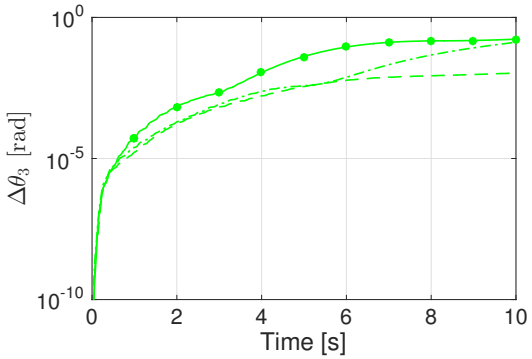
Figure 5.8: Simulation results for the Canadarm example.



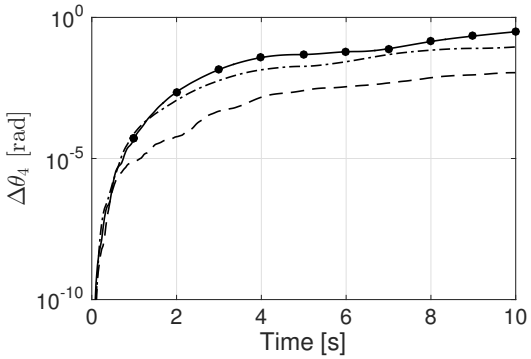
(a) Approximation Error for θ_1



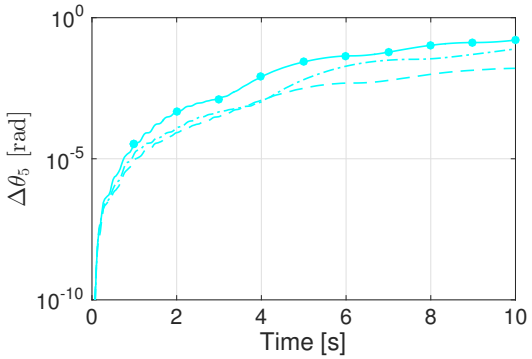
(b) Approximation Error for θ_2



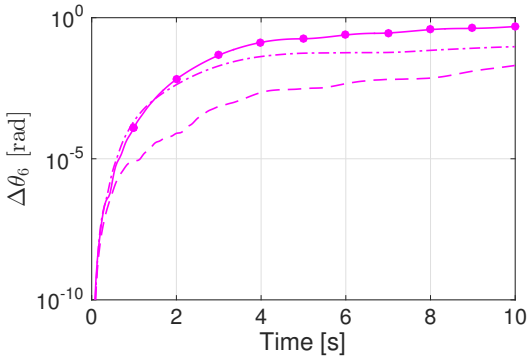
(c) Approximation Error for θ_3



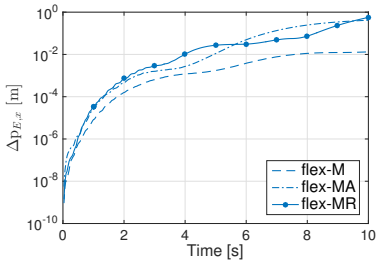
(d) Approximation Error for θ_4



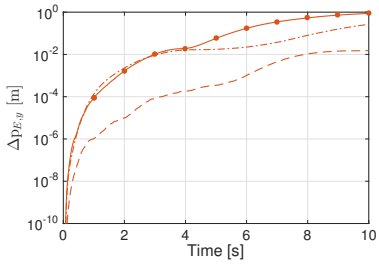
(e) Approximation Error for θ_5



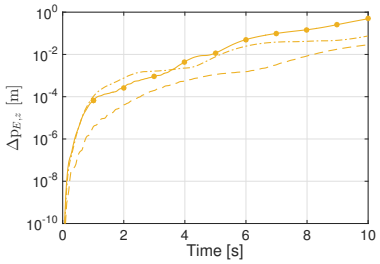
(f) Approximation Error for θ_6



(g) Approximation Error for $\mathbf{p}_{E,x}$



(h) Approximation Error for $\mathbf{p}_{E,y}$



(i) Approximation Error for $\mathbf{p}_{E,z}$

Figure 5.9: Approximation errors for the Canadarm example.

Table 5.1: RMS energy drift and computation time ratios for both examples

Model	RMS error		Time Ratio	
	2DoF	6DoF	2DoF	6DoF
`flex`	$2.67 \cdot 10^{-7}$	$2.81 \cdot 10^{-5}$	100 %	100 %
`flex-M`	$2.63 \cdot 10^{-7}$	$4.48 \cdot 10^{-5}$	51.7%	64.5%
`flex-MA`	$3.93 \cdot 10^{-4}$	$2.09 \cdot 10^{-3}$	49.8%	60.7%
`flex-MR`	$2.49 \cdot 10^{-4}$	$1.43 \cdot 10^{-2}$	44.7%	63.5%

no data were provided about the computation of flexible modes in (Mohan and Saha, 2009).

The second illustration of the inverse dynamics performance is brought by the RMS indicator introduced above, based on (Damaren and Sharf, 1995). Its value must stay as low as possible to confirm that numerical errors are negligible. The results obtained for the two examples are summarized in Table 5.1. Simulations are performed on a processor Intel Core i7 (2.3GHz, 16Go of RAM) with the software Matlab/Simulink©, and using a Runge-Kutta integration scheme of 4^{th} order with a time step of 0.001 s. It is important to highlight that this RMS error is independent of the time step chosen, because it is inherent to the approximations made in the model. In the case of a rigid body, this error drops to 10^{-14} and can be almost lowered to the machine epsilon by reducing the time step. By contrast, some hypothesis are made for the full flexible dynamics, such as the small angle hypothesis for the angular flexible deformations, and they prevent this error from decreasing with the time step. This latter thus represents a good indicator of the model accuracy.

• **Approximation of Flexible Dynamics**: The three test cases `flex-M`, `flex-MA` and `flex-MR` yield varying approximation errors compared to the reference case `flex`. The RMS indicators are provided for each model in Table 5.1. It is clearly seen in the free dynamics of the planar case that this error remains the lowest for the mass matrix approximation `flex-M`. It means that the corrective terms on inertia are not the most important ones in flexible dynamics. As soon as an approximation is made on the lever arm $\mathbf{p}_{i-1}$ in the matrix $\mathbf{A}_{i,i-1}$, this error rises significantly. Indeed, this flexible correction brings out lateral translations on $\mathbf{p}_{i-1}$ that have tremendous dynamic effects due to the arm length. Eventually, the most critical correction comes from the rotation matrices. When neglected, the same phenomenon happens since a flexible rotation at the arm base implies huge motion at the effector level.

The computation time is given in Figure 5.10, where it is seen how the approximate models

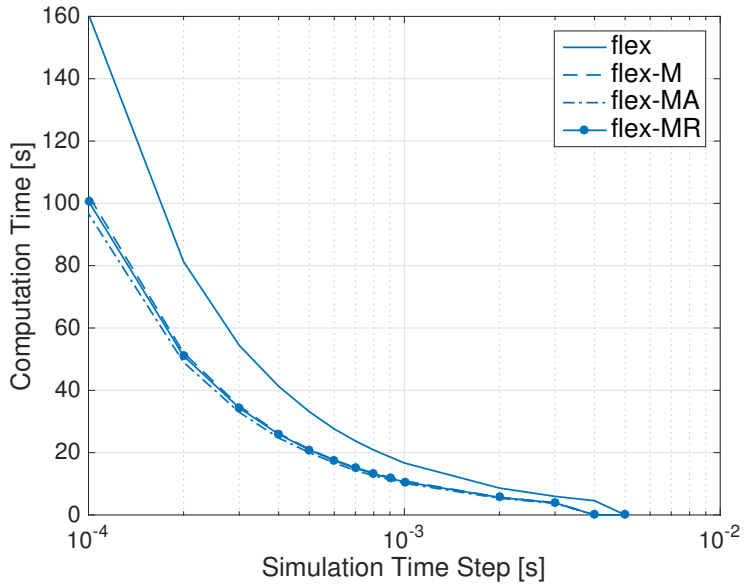


Figure 5.10: Computation time according to simulation time step for the Canadarm example.

reduce it drastically. According to Table 5.1, they nearly reach 40% of computation time reduction, but the `flex-M` model is the only one to keep the RMS error as small as the full flexible model. Indeed, the last two models `flex-MA` or `flex-MR` are not worth it because the computation time is nearly the same as the `flex-M` one, but the resulting RMS error soar. Therefore, this model seems to be the best compromise to reduce the computation time while keeping the error small enough.

5.2.7 Conclusion

A nonlinear model has been presented through this whole chapter to perform efficient and still accurate simulations of flexible manipulators. Thanks to a partitioning of main kinematic and dynamic terms according to their dependency on the flexible coordinates, approximate models are built to alleviate the computational load while keeping the induced error as small as possible.

The proposed generic model of a flexible segment accounts for traction and torsion along the main axis, and bending in its both transverse directions. The nonlinear dynamics have been developed by constantly decomposing the resulting matrices into a nominal value and its corrective terms. These latter are due to changes in the segment shape with the flexible deformation, and result in first and second order terms w.r.t. the flexible coordinates. Using

the DeNOC approach, a Newton-Euler inverse dynamics scheme is obtained under matrix form instead of traditional integral expressions. These results are a key point in order to speed up simulations and avoid on-line evaluation of integrals. Finally, four different models have been tested in simulation in free and forced dynamics. A reference case considering all the nonlinearities was used for high-fidelity simulations, while three approximate models were used to alleviate the computational burden. A first one neglects the corrective terms at the inertial level, impacting both the mass matrix and the Coriolis and centrifugal terms. The second and third models neglect the kinematic corrections on, respectively, the end-tip translation and rotation of the flexible segments. It came out that the less significant corrective terms are the inertial ones, while the flexible translations and rotations at the end-tip are crucial and have a greater influence on global dynamics because of the lever arm effect.

As a conclusion, it is shown that the inertial approximate model `flex-M` is the best compromise to keep the accuracy high enough, while lowering substantially the computational load, and thus, the simulation time.

CHAPTER 6 SPACE ROBOT MODELING

The framework to model fixed-base robotic systems with the DeNOC approach is now extended to include the base dynamics. In the scope of a space robot, the base is moving freely in the terrestrial gravity field and obey the orbital mechanics. The parameterization of the Earth orbits is presented first through the Keplerian elements introduced in the literature review. Then the manipulator models are extended to encompass the base motion under external efforts. Eventually, the orbital and multi-body dynamics of the space robot are decoupled to describe the capture phase in conditions of weightlessness.

6.1 Spacecraft Modeling

The modeling and dynamics of a usual spacecraft are covered. This description highlights the elements that allow the satellite to control its attitude and orbit, as well as its thermal state, among others. Based on the orbital mechanics, these inputs maintain the satellite on the required orbit for the mission by counteracting the drift effects due to the on-orbit disturbances.

6.1.1 Satellite components

As described in the literature review and illustrated in Figure 6.1, a common spacecraft is made of 7 key elements to fulfill a mission. In this thesis, the main focus is put on the design of the *Attitude Control System (ACS)* of a space robot, but most of the following items would be affected and re-designed to embed a manipulator on-board. One may classify them as follows:

- **Structure:** this term represents all the mechanical elements that make up the satellite. They are designed to reach the best compromise between mass and rigidity, while maintaining the spacecraft integrity throughout its lifetime. For example, it must endure high vibrations at launch, and a harsh space environment during the mission, with radiations and large temperature variations.
- **TCS:** the *Thermal Control System (TCS)* regulates the temperature of the whole spacecraft to maintain the equipments in their nominal range of operation, especially mechanical elements and electronic devices. It dissipates the high temperatures encountered when facing the Sun, or heats the spacecraft when passing through the shadow of the Earth, with very low temperatures.

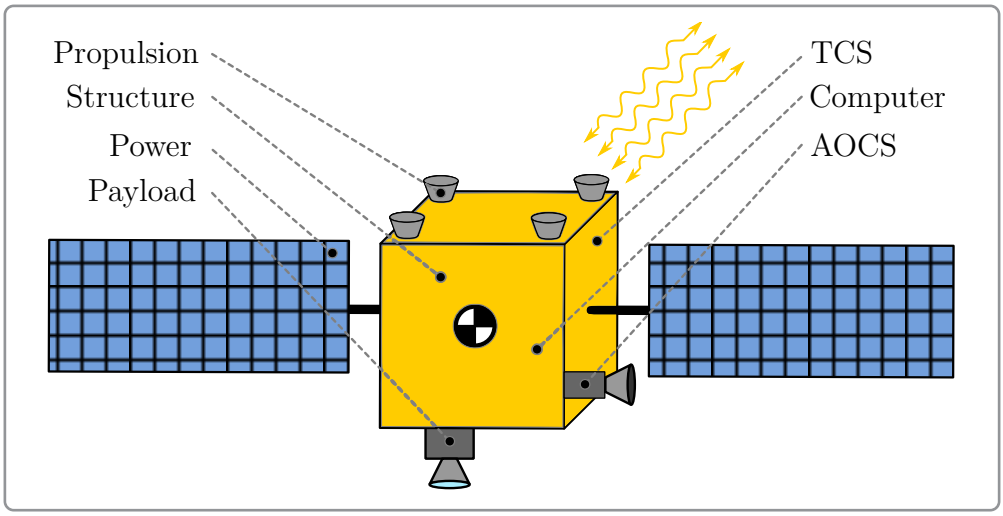


Figure 6.1: Usual components of a spacecraft

- **Propulsion:** the thrusters apply forces and torques on the spacecraft to move it on the desired orbit and to rotate it, as required by the AOCS. The limited amount of fuel on board the spacecraft is the most crucial factor for its operating life.
- **Power:** this system generates all the necessary electrical power to the on-board computer, the AOCS and the payload. It is based on electrical generators, like solar arrays.
- **AOCS:** thanks to actuators, like thrusters, and to sensors, like *Inertial Measurement Unit (IMU)* and star trackers, the AOCS maintains the spacecraft orbit and attitude to fulfill the mission requirements. It can point accurately toward the Earth for communication, or head the solar panels toward the Sun for electricity generation.
- **On-board Computer:** this element is the hub of the spacecraft. It gathers all data and measurements to check the nominal functioning of each equipment, and then send the commands required by the AOCS.
- **Payload:** this last item depends on the purpose of the mission. It can be a camera, a radar or an embedded experiment to perform on-orbit.

The mission goal usually implies that the spacecraft follows a given trajectory with a desired attitude. These requirements are based on the so-called *orbital mechanics*, which describes the system behavior in the terrestrial gravity field.

6.1.2 Orbital mechanics

The spacecraft orbital dynamics is described as a rigid body orbiting the Earth (Hughes, 1986; Chobotov, 1991). The orbit is commonly described by the six *Keplerian elements* illustrated in Figure 6.2. They were mentioned in the literature review, and can be classified

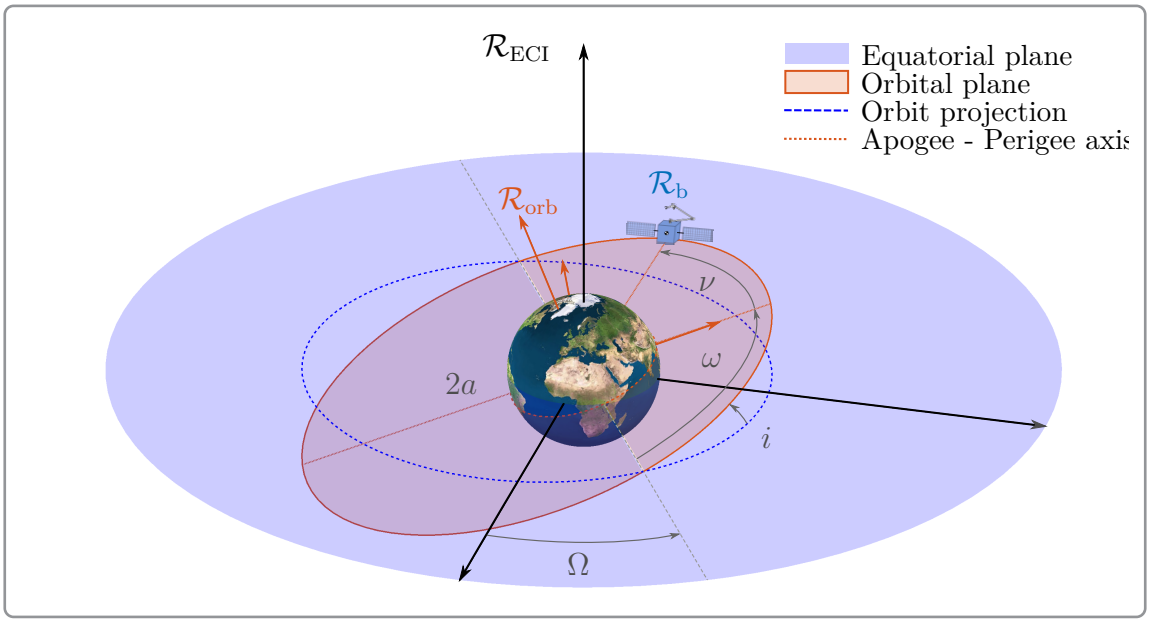


Figure 6.2: Keplerian elements of the elliptical orbit

according to the quantity that they describe:

- **Orbital plane:** the Right Ascension of the Ascending Node (RAAN) Ω, the inclination i, and the argument of the perigee ω ;
- **Shape of the orbit:** the eccentricity e and the semi-major axis a ;
- **Satellite position:** either the mean anomaly M, the eccentric anomaly E, or the true anomaly ν.

It is worth highlighting that the orbit followed by a satellite is a periodic trajectory in the absence of external efforts (i.e., without manœuvers or disturbances). Therefore, the first two categories are used to describe the position and orientation of the orbit in space, while the last one is used to parameterize the spacecraft location. The five first parameters are thus constant in time, while the last one is the only time-variant parameter.

Frames: Three main frames are considered to describe the motion of a spacecraft orbiting the Earth: the *Earth-Centered Inertial (ECI)* frame $\mathcal{R}_{\text{ECI}}$, the inertial frame $\mathcal{R}_{\text{orb}}$ fixed to the orbital plane, and the body-fixed frame $\mathcal{R}_{\text{b}}$. The ECI frame is assumed inertial and located at the Earth center. An equivalent frame may also be defined at the same location, but rotating with the Earth around its own axis: *Earth-Centered-Earth-Fixed (ECEF)* frame is required to manage the communications with ground stations or for mission purposes, like the visibility of a given land or ocean by the payload. The frame $\mathcal{R}_{\text{orb}}$ is not commonly used in space engineering but it is introduced here for kinematics considerations. It is also

fixed at the Earth center and is aligned with the principal axes of the ellipse describing the orbit. Finally, the body-fixed frame is denoted $\mathcal{R}_\mathrm{b}$ and describe the inertial attitude of the spacecraft.

Free-flying dynamics: The spacecraft is considered as a rigid body moving in the gravity field. Its dynamics is governed by the single rigid segment model in (4.20), expressed as follows at the CoM G_b:

$$\boldsymbol{w}_\mathrm{c} + \boldsymbol{w}_\mathrm{e} = \mathbf{M}_\mathrm{b}\dot{\mathbf{t}}_\mathrm{b} + \boldsymbol{\Omega}_\mathrm{b}\mathbf{M}_\mathrm{b}\mathbf{E}_v\mathbf{t}_\mathrm{b} \quad \Longrightarrow \quad \left\{ \begin{array}{ll} \boldsymbol{f}_\mathrm{c} + \boldsymbol{f}_\mathrm{e} = m_\mathrm{b}\,\dot{\boldsymbol{v}}_\mathrm{b} & \text{(6.1a)} \\ \boldsymbol{n}_\mathrm{c} + \boldsymbol{n}_\mathrm{e} = \mathbf{I}_\mathrm{b}\,\dot{\boldsymbol{\omega}}_\mathrm{b} + \boldsymbol{\omega}_\mathrm{b} \times (\mathbf{I}_\mathrm{b}\boldsymbol{\omega}_\mathrm{b}) & \text{(6.1b)} \end{array} \right.$$

with $\mathbf{t}_\mathrm{b} = \left[\begin{array}{cc} \boldsymbol{v}_\mathrm{b}^\top & \boldsymbol{\omega}_\mathrm{b}^\top \end{array} \right]^\top$ the spacecraft twist defined by $\boldsymbol{v}_\mathrm{b} = \boldsymbol{v}(G_\mathrm{b})$, and $\mathbf{M}_\mathrm{b} = \mathrm{diag}(m_\mathrm{b}, \mathbf{I}_\mathrm{b})$ its mass matrix at G_b. The efforts are classified according to the control ones, coming from the AOCS with $\boldsymbol{w}_\mathrm{c} = (\boldsymbol{f}_\mathrm{c}, \boldsymbol{n}_\mathrm{c})$, and to the external ones due to the environment with $\boldsymbol{w}_\mathrm{e} = (\boldsymbol{f}_\mathrm{e}, \boldsymbol{n}_\mathrm{e})$. The net external force $\boldsymbol{f}_\mathrm{e}$ is the system weight $\boldsymbol{f}_\mathrm{g}$ due to the terrestrial attraction, while the net external torque $\boldsymbol{n}_\mathrm{e}$ reduces to the so-called *gravity gradient*[1] $\boldsymbol{n}_\mathrm{g}$. They read as follows when expressed at G_b ((Hughes, 1986),p.236):

$$\boldsymbol{f}_\mathrm{g} = \frac{\mu\, m_\mathrm{b}}{r_\mathrm{b}^2}\boldsymbol{u}_\mathrm{b} \tag{6.2a}$$

$$\boldsymbol{n}_\mathrm{g} = 3\,\frac{\mu}{r_\mathrm{b}^3}\,\boldsymbol{u}_\mathrm{b} \times (\mathbf{I}_{G_\mathrm{b}}\boldsymbol{u}_\mathrm{b}) \tag{6.2b}$$

where $\mu = \mathcal{G}\, m_\mathrm{earth}$, with $\mathcal{G} = 6.67 \times 10^{-11} N.m^2/kg^2$ the universal gravitational constant, and m_earth the mass of the Earth. The inertial position of the spacecraft from $\mathcal{R}_\mathrm{ECI}$ is given by $\mathbf{r}_\mathrm{b} = \overrightarrow{O_\mathrm{ECI}G_\mathrm{b}}$, and the corresponding unitary vector is denoted by $\boldsymbol{u}_\mathrm{b} = -\mathbf{r}_\mathrm{b}/r_\mathrm{b}$. Vectors are denoted in bold characters while their norm is denoted by the same character in regular weight, i.e., $r_\mathrm{b} = ||\mathbf{r}_\mathrm{b}||$.

Inertial position of the spacecraft: The linear dynamic equation can be easily integrated twice and provide the inertial position of the spacecraft CoM in space. This periodic trajectory is called the "orbit", and can result in an ellipse, a hyperbola or a parabola. Focusing on Earth satellites, only ellipses are considered in the sequel. They are fully described by the Keplerian elements mentioned above, where (Ω, i, ω) are a set of Euler angles defining

[1]This torque is due to the non-homogeneous distribution of the differential weight: $\mathrm{d}\boldsymbol{f}_\mathrm{g} = \mu\,\mathrm{d}m/r_M^2\boldsymbol{u}$ on an element $\mathrm{d}m$ of the spacecraft, located at a distance r_M from the Earth's center O_ECI, along the unitary vector $\boldsymbol{u}_M = -\mathbf{r}_M/r_M$ with $\mathbf{r}_M = \overrightarrow{O_\mathrm{ECI}M}$. Since each differential element $\mathrm{d}m$ is located at a slightly different distance and along a different vector, the resulting torque is non-zero at the spacecraft's CoM. It is worth mentioning that the torque expression in (6.2b) is only valid at the CoM, otherwise an extra term appears to account for the lever arm with the new reference point (see (Hughes, 1986), p.247).

the orbital plane orientation, (e, a) are the ellipse parameters, and ν is the polar parameterization of the spacecraft location. Instead of using the semi-major axis a, orbits are often described by their altitude h at the perigee, which is the closest point from the Earth. To summarize, the following set of six parameters is required to fully characterize an object on a terrestrial orbit:

Ω	[rad]	RAAN
i	[rad]	Inclination
ω	[rad]	Argument of the perigee
h	[m]	Altitude of the perigee
e	[-]	Eccentricity
ν	[rad]	True anomaly

The orbital plane is obtained by a rotation from $\mathcal{R}_{\mathrm{ECI}}$, with its X-axis pointing toward the perigee:

$$^{\mathrm{I}}R_{\mathrm{orb}} = \mathrm{Rot}(\Omega, Z)\,\mathrm{Rot}(i, X)\,\mathrm{Rot}(\omega, X) \tag{6.3}$$

Then, the mathematical description of the ellipse, in the orbital plane, is based on the eccentricity e and the *semi-latus rectum* l, deriving from the semi-major axis a, itself deriving from the altitude h, as follows:

$$a = (R_{\mathrm{earth}} + h) \,/\, (1 - e)$$

$$l = a\,(1 - e^2)$$

with R_{earth} the radius of the Earth. With these two parameters, the polar parameterization of the ellipse (r_{b}, ν) in the orbital plane reads:

$$r_{\mathrm{b}}(\nu) = \frac{l}{1 + e\,\cos(\nu)} \tag{6.4}$$

yielding the cartesian position of the spacecraft in the orbital plane by:

$$^{\mathrm{orb}}\mathbf{r}_{\mathrm{b}}(\nu) = \begin{bmatrix} r_{\mathrm{b}}(\nu)\,\cos(\nu) \\ r_{\mathrm{b}}(\nu)\,\sin(\nu) \\ 0 \end{bmatrix} \tag{6.5}$$

Finally, applying the rotation from $\mathcal{R}_{\mathrm{ECI}}$, its inertial position is obtained by:

$$\mathbf{r}_{\mathrm{b}}(\nu) = {}^{\mathrm{I}}R_{\mathrm{orb}}\,{}^{\mathrm{orb}}\mathbf{r}_{\mathrm{b}}(\nu) \tag{6.6}$$

Inertial attitude of the spacecraft: As opposed to linear dynamics, angular ones cannot be obtained analytically, but requires instead a numerical integration to obtain the angular rate $\boldsymbol{\omega}_\mathrm{b}$ from (6.1b). Depending on the frame of expression of $\boldsymbol{\omega}_\mathrm{b}$, the transformation matrix $R_{\boldsymbol{\Psi}_\mathrm{b}}$ or $R_{\mathcal{Q}_\mathrm{b}}$ in Appendix A.2 is used to obtain the time derivative of the Euler angles $\boldsymbol{\Psi}_\mathrm{b}$ or of the quaternion $\mathcal{Q}_\mathrm{b}$ describing the spacecraft attitude. Integrating this last quantity, the spacecraft orientation is obtain w.r.t. the inertial frame $\mathcal{R}_\mathrm{ECI}$, which does not account for the Earth rotation on its own axis.

6.1.3 External disturbances

On the previous dynamic model (6.2), some usual disturbances are taken into account in the forces and torques applied on the spacecraft. The main sources of disturbance on LEO are coming from the irregular shape of the Earth, from the lunar and solar influences, from the higher layers of the atmosphere, and from the solar radiation. They are introduced for sake of completeness and will be treated as general disturbances in the simulation scheme of the thesis. If one needs to model them more accurately, their influence on the previous model is given as follows, by order of magnitude from the most influential to the least (Escudier and Pouillard, 1996):

$$\boldsymbol{f}_\mathrm{e} = \boldsymbol{f}_{\mathrm{g},earth} + \boldsymbol{f}_\mathrm{drag} + \boldsymbol{f}_\mathrm{g,moon} + \boldsymbol{f}_\mathrm{g,sun} + \boldsymbol{f}_\mathrm{pressure} \tag{6.7}$$

with $\boldsymbol{f}_\mathrm{g}$ the gravitational forces coming from the Earth, the Moon and the Sun, $\boldsymbol{f}_\mathrm{drag}$ the atmospheric drag, and $\boldsymbol{f}_\mathrm{pressure}$ the solar-radiation pressure. Among them, only the drag, the pressure and the terrestrial gravity would induce a torque. The forces are detailed in the sequel to have a glimpse into the way disturbances are considered, but the torques are not expressed since they require additional integrations on the whole body or on its exposed surface and would not bring much contribution to the present discussion. The reader can refer to (Hughes, 1986) to find a thorough derivation and analysis of these disturbances.

Earth's Gravity: The attractive force given in (6.2a) derives from the terrestrial potential of gravity for a spherical body. The irregularities on the Earth surface nullify this hypothesis, and result in corrective terms on this potential. Many terms could be introduced to describe it more faithfully, but the main ones are the spherical harmonics which assume a symmetry of revolution. For example, the first zonal harmonic J_2 is the most important gravitational correction, and depicts the Earth flattening.

The gravity potential used in (6.2) is based on the perfect sphere hypothesis, and reads:

$$U(r) = \frac{\mu}{r}$$

while the first zonal harmonic J_2 corrects it with an extra term:

$$U(\mathbf{r}) = \frac{\mu}{r} + \frac{\mu}{r}\left(\frac{R_{\text{earth}}}{r}\right)^2 J_2 \frac{3\sin^2(\phi) - 1}{2}$$

with ψ the latitude of the spacecraft projection on the Earth surface. The general potential is actually obtained by integrating it over the entire Earth, leading to a spherical development in zonal and tesseral harmonics (Escudier and Pouillard, 1996):

$$U(\mathbf{r}) = \frac{\mu}{r} + \sum_{n=1}^{\infty}\left(\frac{r_{eq}}{r}\right)^n \left[-J_n P_n(\sin(\phi)) + \sum_{m=1}^{n}\{C_{n,m}\cos(m\,\lambda) + S_{n,m}\sin(m\,\lambda)\}P_{n,m}(\sin(\phi))\right]$$

with (ϕ, λ) the latitude/longitude of the spacecraft projection on the Earth surface, J_n the zonal harmonics and $(C_{n,m}, S_{n,m})$ the tesseral ones. P_n denote the Legendre polynomials, and $P_{n,m}$ the corresponding Legendre functions.

This new potential is then derived with the gradient operator ∇ to correct the gravity force applied on the spacecraft by:

$$\boldsymbol{f}_{\text{g,earth}} = -m_{\text{b}}\,\nabla U(\mathbf{r})$$

Considering the induced torque, this gravity force must be integrated on the whole body to obtain the resulting gravity gradient $\boldsymbol{n}_{\text{g}}$. The interested reader is referred to ((Hughes, 1986),p.236) to find the corresponding methodology.

Atmospheric Drag: The atmosphere density is globally decreasing with the altitude, but the remaining particles in its higher layers are sufficiently dense to create a drag force on any object in motion. Usually, this disturbance is worth considering up to an altitude of 1,500 km. Aerodynamics theory yields the following drag force ((Hughes, 1986), p258):

$$\boldsymbol{f}_{\text{drag}} = -\frac{1}{2}\,\rho\,S\,v_r^2\,C_D\,(\boldsymbol{n}^\top \boldsymbol{u}_v)\,\boldsymbol{u}_v$$

with

ρ	$[kg/m^3]$	Air density at the current altitude
S	$[m^2]$	Reference surface
$\boldsymbol{v}_r$	$[m/s]$	Relative speed w.r.t. the atmosphere
$\boldsymbol{n}$	$[-]$	Unitary outward normal to the surface
$\boldsymbol{u}_v$	$[-]$	Unitary vector along the relative speed, ($\boldsymbol{u}_v \triangleq \boldsymbol{v}_r/v_r$)
C_D	$[-]$	Drag coefficient

The scalar product between both unitary vectors accounts for the "effective" surface seen from the air flow. This disturbance is significant because of the spacecraft velocity, since it can reach 10 km/s in LEO.

Moon/Sun's Gravity: The third main disturbance is the gravitational influence of other bodies on the spacecraft trajectory. Around the Earth, the Moon is accounted for because of its closeness, and the Sun for its considerable mass compared to other objects in the solar system. Their effect is given by the addition of gravity forces acting on the spacecraft[2], as follows:

$$\boldsymbol{f}_{\text{g,moon}} = -\frac{\mu_{\text{moon}}\, m_{\text{b}}}{r_{O_M G_{\text{b}}}} \qquad\qquad \boldsymbol{f}_{\text{g,sun}} = -\frac{\mu_{\text{sun}}\, m_{\text{b}}}{r_{O_S G_{\text{b}}}}$$

denoting by $\mu_{\text{moon}} = \mathcal{G}\, m_{\text{moon}}$ and $\mu_{\text{sun}} = \mathcal{G}\, m_{\text{sun}}$ the gravitational constant of, respectively, the Moon and the Sun, and by $r_{O_i G_{\text{b}}}$ the distance of the spacecraft from the CoM of these two bodies.

Solar Pressure: The least influential disturbance is the pressure induced by the solar radiations. The flux of photons coming from the Sun produces a force acting on every exposed surface of the spacecraft. Once received by these surfaces, it is partially absorbed and reflected. Two different types of reflection take place: the "diffuse" one and the "specular" one. Each of them is emitted in a specific direction: the diffuse reflection is along the surface normal vector, while the specular reflection obey the Snell-Descartes laws of reflection.

To summarize, each surface is subject to a pressure due to the total amount of photons reaching it, and two additional pressures due to the fraction emitted along the normal direction (i.e., diffuse) and to the one emitted along the symmetric direction (i.e., specular). The thorough derivations of forces and moments induced by the solar pressure are available in (Hughes, 1986)(p.263), and are only mentioned here for completeness. The force may be written as follows for an homogeneous and plane surface:

$$\boldsymbol{f}_{\text{pressure}} = -P_0 \left(\boldsymbol{n}^\top \boldsymbol{i}\right) \left[(\sigma_a + \sigma_{rd})\, \boldsymbol{i} + \frac{2}{3}\, \sigma_{rd}\, \boldsymbol{n} + 2\, \sigma_{rs}\, \boldsymbol{i}_s\right]$$

with

[2]It is worth mentioning that the term "disturbance" is well-suited here. Indeed, the idea to add the effect of these bodies only as external forces is valid for a short amount of time, and hide a much more complex reality. Since the Earth is rotating the Sun, the frame $\mathcal{R}_{\text{ECI}}$ is no longer inertial in the solar system. Hence, the spacecraft dynamics for long term considerations should be entirely re-develop with respect to the Sun-Centered frame, which is assumed inertial for the solar system.

P_0	$[N/m^2]$	Solar radiation pressure at 1 A.U.
S	$[m^2]$	Surface
$\boldsymbol{n}$	$[-]$	Unitary outward normal vector of the surface
$\boldsymbol{i}$	$[-]$	Unitary Sun-pointing vector
$\boldsymbol{i}_s$	$[-]$	Reflected vector of $\boldsymbol{i}$ w.r.t. $\boldsymbol{n}$
σ_a	$[-]$	Absorption coefficient
σ_{rd}	$[-]$	Diffuse reflexion coefficient
σ_{rs}	$[-]$	Specular reflexion coefficient

The first scalar product $\left(\boldsymbol{n}^\top \boldsymbol{i}\right)$ is used to compute the "effective" surface seen from the flux of photons, and the term 2/3 in the diffuse reflection accounts for thermodynamics considerations.

The dynamic model of a spacecraft orbiting the Earth has been quickly covered through the previous considerations. It was considered as a single rigid body subject to external and control efforts. The control inputs are mostly used to place the satellite onto its desired orbit after its release from the launcher, and to counteract the disturbances raised above. The dynamic coupling of a spacecraft with an embedded robotic arm and other appendages is now investigated to represent the behavior of a space robot in orbit.

6.2 Space Robot Modeling

The term "space robot" refers to a rigid hub, called *spacecraft*, connected to a set of various *appendages*, either rigid or flexible, and either actuated or fixed. Thanks to the dynamic models for rigid bodies in (4.20) and for flexible ones in (5.30), global dynamics of a multi-body system mixing rigid and flexible elements is derived hereunder. A first part introduces the additional notations and equations at the crossing of spacecraft dynamics and robotics. Then, the resulting algorithms to compute the inverse and forward dynamics are presented, before highlighting the main hypotheses made on these dynamics for the capture scenario.

6.2.1 Dynamic coupling

The coupling between the hub and its appendages is investigated by updating the common steps of modeling, through kinematics, kinetics and dynamics. The focus is placed on the robotics point of view rather than on the orbital mechanics one, since it allows to include an additional body to the chain with less efforts. The development of the space robot model is mainly derived from the humanoid robot modeling based on the DeNOC approach in (Shah et al., 2012a; Saha et al., 2013).

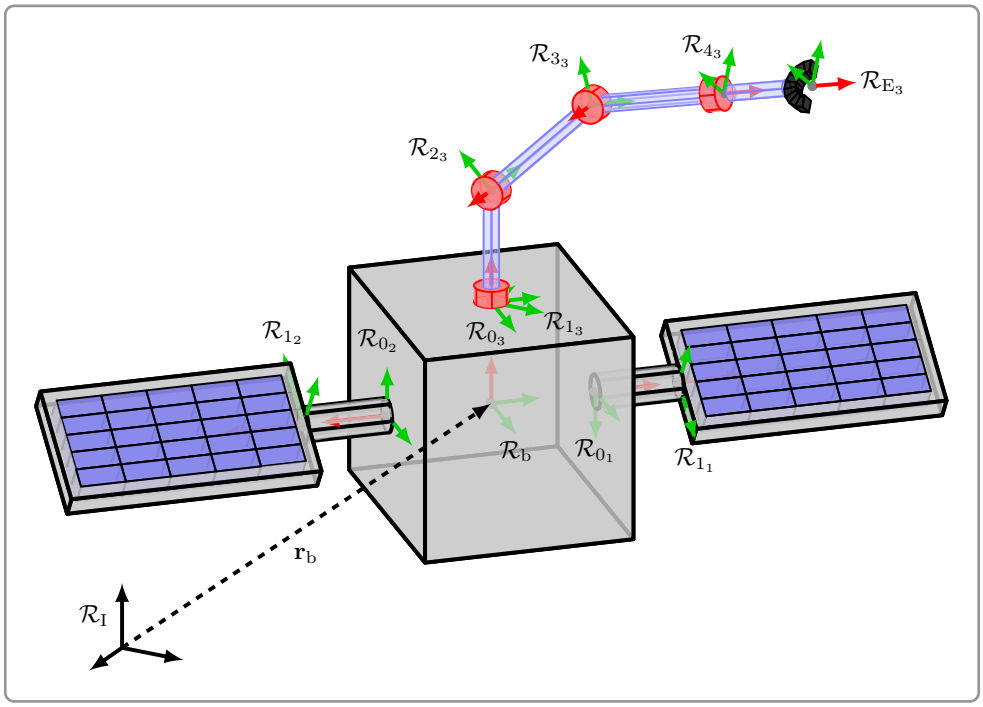


Figure 6.3: Frames assignment on a spacecraft with various appendages

6.2.1.1 Kinematics

Kinematics is based on the frames illustrated in Figure 6.3. The space robot has n_a appendages, and the k^{th} appendage has n_k bodies. Inside the k^{th} appendage, the frame $\mathcal{R}_{i_k}$ denotes the reference frame of the i^{th} body. It is worth noticing that some appendages have only one element, as the solar panels or the reaction wheels, while other exhibit many sub-elements, as the robotic arms. The notations introduced earlier are thus updated by the appendage index, by denoting the unconstrained coordinates of any body as follows:

$$\mathbf{x}_{i_k} = \begin{bmatrix} \mathbf{r}_{i_k}^\top & \mathbf{\Psi}_{i_k}^\top & \boldsymbol{\delta}_{i_k} \end{bmatrix}^\top \tag{6.8}$$

where the index "i_k" allows to keep using the same notations as before, describing thus each appendage as a robotic arm. The position and the rotation matrix of each body can be decomposed according to the base coordinates and to the relative position of the anchorage point of the appendage, as follows:

$$\mathbf{r}_{i_k} = \mathbf{r}_b + {}^b\mathbf{r}_{0_k} + {}^{0_k}\mathbf{r}_{i_k}(\mathbf{q}_{a_k}) \tag{6.9a}$$

$${}^I R_{i_k} = {}^I R_b(\mathbf{\Psi}_b)\, {}^b R_{0_k}\, {}^{0_k} R_{i_k}(\mathbf{q}_{a_k}) \tag{6.9b}$$

where $\mathbf{q}_{a_k}$ denotes the generalized coordinates of the k^{th} appendage. The upper left-hand

side superscripts denote the frame of reference from which the positions or the rotations are expressed.

In the previous relations, the parameters $\left({}^{\mathrm{b}}\mathbf{r}_{0_k}, {}^{\mathrm{b}}R_{0_k}\right)$ are constant and inherent to the location and orientation of the appendage basis on the spacecraft, as illustrated in Figure 6.3. They are considered as model parameters in the sequel.

On the other hand, the pair $({}^{0_k}\mathbf{r}_{i_k}, {}^{0_k}R_{i_k})$ is a function of $\mathbf{q}_{\mathrm{a}_k}$, given by the kinematic model of the appendage, by (4.25) for the rigid case, and by (5.38) for the flexible one. They are obtained by the appendage kinematic transformation ${}^{0_k}T_{i_k}$ with ${}^{0_k}\mathbf{r}_{i_k} = {}^{0_k}T_{i_k}(1{:}3, 4)$ and ${}^{0_k}R_{i_k} = {}^{0_k}T_{i_k}(1{:}3, 1{:}3)$. These transformation are defined by the appendage generalized coordinates $\mathbf{q}_{\mathrm{a}_k}$, while the generalized coordinates of the spacecraft are unconstrained, and are denoted by:

$$\mathbf{x}_{\mathrm{b}} = \begin{bmatrix} \mathbf{r}_{\mathrm{b}}^{\top} & \mathbf{\Psi}_{\mathrm{b}}^{\top} \end{bmatrix}^{\top} \tag{6.10}$$

such that the global vector of generalized coordinates is given by merging the spacecraft coordinates $\mathbf{x}_{\mathrm{b}}$ and the appendages coordinates $\mathbf{q}_{\mathrm{a}}$ as follows:

$$\mathbf{q} = \begin{bmatrix} \mathbf{x}_{\mathrm{b}}^{\top} & \mathbf{q}_{\mathrm{a}}^{\top} \end{bmatrix}^{\top} \tag{6.11}$$

$$\mathbf{q}_{\mathrm{a}} = \begin{bmatrix} \mathbf{q}_{\mathrm{a}_1}^{\top} & \dots & \mathbf{q}_{\mathrm{a}_{n_{\mathrm{a}}}}^{\top} \end{bmatrix}^{\top} \tag{6.12}$$

The total number of variables is given by $N = 6 + \sum_{k=1}^{n_{\mathrm{a}}} s_k$, denoting by s_k the numbers of DoF of the whole appendage, and s_{i_k} for its i^{th} body. It yields that $s_k = \sum_{i_k} s_{i_k}$.

As a conclusion, the position and orientation of any body of the space robot is obtained by a relation merging the generalized coordinates of the hub with those of its own appendage:

$$\mathbf{x}_{i_k} = f_{kin}(\mathbf{x}_{\mathrm{b}},\, \mathbf{q}_{\mathrm{a}_k}) \tag{6.13}$$

This relation is mostly applied to the end-effectors of the embedded manipulators. Combining the relative position and orientation w.r.t. its base in (4.26) and (5.39) with (6.9), the end-effectors are described by the following kinetic model, based on the transformation matrix ${}^{0_k}T_{\mathrm{E}_k}(\mathbf{q}_{\mathrm{a}_k})$:

$$\mathbf{r}_{\mathrm{E}_k} = \mathbf{r}_{\mathrm{b}} + {}^{\mathrm{b}}\mathbf{r}_{0_k} + {}^{0_k}\mathbf{r}_{\mathrm{E}_k}(\mathbf{q}_{\mathrm{a}_k}) \tag{6.14a}$$

$${}^{I}R_{\mathrm{E}_k} = {}^{I}R_{\mathrm{b}}(\mathbf{\Psi}_{\mathrm{b}})\, {}^{\mathrm{b}}R_{0_k}\, {}^{0_k}R_{\mathrm{E}_k}(\mathbf{q}_{\mathrm{a}_k}) \tag{6.14b}$$

6.2.1.2 Kinetics

The time derivation of (6.13) provides a recursive relation defining the twists in a similar way as (4.28), except that the initial base twist is no longer $\mathbf{0}_{6\times1}$. Indeed, the anchorage point of the appendage is moving with the spacecraft, whose motion is described by twist $\mathbf{t}_\mathrm{b}$ in (6.1). The twist is chosen as an hybrid coordinate to describe the spacecraft generalized velocities, rather than the time derivative of $\mathbf{x}_\mathrm{b}$ in (6.10); exactly as it was done previously to describe the dynamics of a single rigid body.

In the first place, the kinematics of any body inside the k^{th} appendage is obtained by updating equation (4.28) with the appendage index:

$$\forall k = 1 \dots n_\mathrm{a},\ \forall i = 1 \dots n_k,\quad \boxed{\mathbf{t}_{i_k} = \mathbf{A}_{i_k,i_k-1}\,\mathbf{t}_{i_k-1} + \mathbf{P}_{i_k}\,\dot{\boldsymbol{q}}_{i_k}} \tag{6.15}$$

and the initialization for any appendage is given by the twist propagation along the spacecraft, considered as a rigid body. They are defined as follows:

$$\forall k = 1 \dots n_\mathrm{a},\quad \boxed{\mathbf{t}_{0_k} = \mathbf{A}_{0_k,\mathrm{b}}\,\mathbf{t}_\mathrm{b}} \tag{6.16}$$

with the twist-propagation matrix $\mathbf{A}_{0_k,\mathrm{b}}$ giving the kinematic transformation to the anchorage point of the appendage by:

$$\mathbf{A}^{\mathrm{r}}_{0_k,\mathrm{b}} = \begin{bmatrix} \mathbf{I}_3 & -\left({}^{\mathrm{b}}\mathbf{r}_{0_k}\right)^{\times} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix} \tag{6.17}$$

Again, the notion of frame of expression is essential. Similar to (4.29a) and (4.30a) for the rigid case, vectors ${}^{\mathrm{b}}\mathbf{r}_{0_k}$ are usually expressed in the spacecraft frame $\mathcal{R}_\mathrm{b}$, such that the rotation to the appendage base frame $\mathcal{R}_{0_k}$ is necessary to fully define the initial twist-propagation matrix as:

$$\mathbf{A}_{0_k,\mathrm{b}} = {}^{\mathrm{b}}\boldsymbol{R}^{\top}_{0_k}\,{}^{(\mathrm{b})}\mathbf{A}^{\mathrm{r}}_{0_k,\mathrm{b}} \tag{6.18}$$

with

$${}^{\mathrm{b}}\boldsymbol{R}_{0_k} = \begin{bmatrix} {}^{\mathrm{b}}R_{0_k} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & {}^{\mathrm{b}}R_{0_k} \end{bmatrix} \tag{6.19}$$

It is also important to expand the spacecraft twist according to the chosen generalized velocities. Let us write the twist propagation at the spacecraft level as:

$$\mathbf{t}_{\mathrm{b}} = \mathbf{P}_{\mathrm{b}}\,\dot{\mathbf{q}}_{\mathrm{b}} \tag{6.20}$$

where the matrix $\mathbf{P}_{\mathrm{b}}$ can be defined in two different ways. If one prefers to obtain the spacecraft twist as the dynamics output for its physical meaning, then the following abuse of notation is introduced $\dot{\mathbf{q}}_{\mathrm{b}} = \mathbf{t}_{\mathrm{b}}$ and $\mathbf{P}_{\mathrm{b}} = \mathbf{I}_6$. In contrast, one may prefer to obtain the time derivative of the position and of the Euler angles in (6.10) to integrate them directly. In that case, $\dot{\mathbf{q}}_{\mathrm{b}}$ is replaced by its natural definition $\dot{\mathbf{q}}_{\mathrm{b}} = \dot{\mathbf{x}}_{\mathrm{b}}$, and $\mathbf{P}_{\mathrm{b}} = R_{\mathbf{x}_{\mathrm{b}}}$, which is introduced to transform the time derivatives of Euler angles into an angular rate in (4.5) .

Jacobian Matrix of an Appendage: Still considering each appendage in the light of robotic systems, the Jacobian matrix for the end-effector of any embedded manipulator is derived in the following. It is worth noticing that this kinetic model could also be applied to any kind of appendage. For example, one may want to describe the kinematics and kinetics of an antenna or of a payload instrument.

The recursion in (6.15) is written under matrix form, as done in the rigid modeling of multi-body systems to derive the Jacobian matrix as well. However, the twist of the first element $\mathbf{t}_{1_k}$ is written directly as a function of the spacecraft twist $\mathbf{t}_{\mathrm{b}}$ by replacing $\mathbf{t}_{0_k}$ by (6.16) in the first step of (6.15).

$$\mathbf{t}_{\mathrm{a}_k} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{2_k,1_k} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{3_k,2_k} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{n_k,n_k-1} & \mathbf{0} \end{bmatrix}}_{\triangleq {}^k\mathbf{A}_k} \mathbf{t}_{\mathrm{a}_k} + \underbrace{\begin{bmatrix} \mathbf{P}_{1_k} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{2_k} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_{3_k} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{P}_{n_k} \end{bmatrix}}_{\triangleq {}^k\mathbf{N}_d} \dot{\mathbf{q}}_{\mathrm{a}_k} + \underbrace{\begin{bmatrix} \mathbf{A}_{1_k,\mathrm{b}} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}}_{\triangleq {}^k\mathbf{A}_{\mathrm{b}}} \mathbf{t}_{\mathrm{b}} \tag{6.21}$$

where the transitivity notation of the twist-propagation matrix allows to write at the base level: $\mathbf{A}_{1_k,\mathrm{b}} = \mathbf{A}_{1_k,0_k}\,\mathbf{A}_{0_k,\mathrm{b}}$. The previous relation is then summarized under matrix form as follows:

$$\mathbf{t}_{\mathrm{a}_k} = {}^k\mathbf{A}_k\,\mathbf{t}_{\mathrm{a}_k} + {}^k\mathbf{N}_d\,\dot{\mathbf{q}}_{\mathrm{a}_k} + {}^k\mathbf{A}_{\mathrm{b}}\,\mathbf{t}_{\mathrm{b}}$$

where the twists of the appendage are merged into :

$$\mathbf{t}_{\mathrm{a}_k} = \begin{bmatrix} \mathbf{t}_{1_k}^{\top} & \dots & \mathbf{t}_{n_k}^{\top} \end{bmatrix}^{\top} \tag{6.22}$$

The kinetic relation defining the DeNOC approach is re-written for one appendage, as follows:

$$\mathbf{t}_{\mathrm{a}_k} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{2_k,1_k} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{3_k,1_k} & \mathbf{A}_{3_k,2_k} & \mathbf{I} & \ddots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{A}_{n_k,1_k} & \mathbf{A}_{n_k,2_k} & \cdots & \mathbf{A}_{n_k,n_k-1} & \mathbf{I} \end{bmatrix}}_{=(\mathbf{I}-{}^k\mathbf{A}_k)^{-1}\triangleq {}^k\mathbf{N}_{k,l}} \underbrace{\begin{bmatrix} \mathbf{P}_{1_k} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{2_k} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_{3_k} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_{n_k} \end{bmatrix}}_{\triangleq {}^k\mathbf{N}_d} \dot{\mathbf{q}}_{\mathrm{a}_k} + \underbrace{\begin{bmatrix} \mathbf{A}_{1_k,\mathrm{b}} \\ \mathbf{A}_{2_k,\mathrm{b}} \\ \mathbf{A}_{3_k,\mathrm{b}} \\ \vdots \\ \mathbf{A}_{n_k,\mathrm{b}} \end{bmatrix}}_{={}^k\mathbf{N}_{k,l}\,{}^k\mathbf{A}_{\mathrm{b}}\triangleq {}^k\mathbf{N}_{\mathrm{b},l}} \mathbf{P}_{\mathrm{b}}\,\dot{\mathbf{q}}_{\mathrm{b}} \tag{6.23}$$

The following notations are then introduced to denote the kinetic relation of the k^{th} appendage with the orthogonal complements of its own influence by ${}^k\mathbf{N}_k$ and of the base influence by ${}^k\mathbf{N}_{\mathrm{b}}$. The kinetic relation is re-written in a more compact form as:

$$\mathbf{t}_{\mathrm{a}_k} = \underbrace{{}^k\mathbf{N}_{k,l}\,{}^k\mathbf{N}_d}_{\triangleq {}^k\mathbf{N}_k}\,\dot{\mathbf{q}}_{\mathrm{a}_k} + \underbrace{{}^k\mathbf{N}_{\mathrm{b},l}\,{}^{\mathrm{b}}\mathbf{N}_d}_{\triangleq {}^k\mathbf{N}_{\mathrm{b}}}\,\dot{\mathbf{q}}_{\mathrm{b}} \tag{6.24}$$

where the matrix $\mathbf{P}_{\mathrm{b}}$ is denoted by ${}^{\mathrm{b}}\mathbf{N}_d$ for coherency with the appendage notations.

One row of this relation allows to define the Jacobian matrix of any body of the appendage by:

$$\mathbf{t}_{i_k} = \left[\begin{array}{ccc|c|ccc} \mathbf{A}_{i_k,1_k} & \cdots & \mathbf{A}_{i_k,i_k-1} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \end{array}\right] {}^k\mathbf{N}_d\,\dot{\mathbf{q}}_{\mathrm{a}_k} + \mathbf{A}_{i_k,\mathrm{b}}\,{}^{\mathrm{b}}\mathbf{N}_d\,\dot{\mathbf{q}}_{\mathrm{b}}$$

which leads to the Jacobian matrix of the end-effector of the k^{th} appendage, by using the twist-propagation matrix for the tool $\mathbf{A}_{n_k+1,n_k}$ introduced in (4.33):

$$\boxed{\mathbf{t}_{\mathrm{E}_k} = \underbrace{\begin{bmatrix} \mathbf{A}_{n_k+1,1} & \mathbf{A}_{n_k+1,2_k} & \cdots & \mathbf{A}_{n_k+1,n_k} \end{bmatrix} {}^k\mathbf{N}_d}_{\triangleq {}^k\mathbf{J}_{\mathrm{E}_k}}\,\dot{\mathbf{q}}_{\mathrm{a}_k} + \underbrace{\begin{bmatrix} \mathbf{A}_{n_k+1,\mathrm{b}} \end{bmatrix} {}^{\mathrm{b}}\mathbf{N}_d}_{\triangleq {}^{\mathrm{b}}\mathbf{J}_{\mathrm{E}_k}}\,\dot{\mathbf{q}}_{\mathrm{b}}} \tag{6.25}$$

This relation is the kinetic equivalent of the relation (6.14). The Jacobian matrix ${}^k\mathbf{J}_{\mathrm{E}_k}$ describes the effect of the joint motion on the effector, while ${}^{\mathrm{b}}\mathbf{J}_{\mathrm{E}_k}$ includes the spacecraft influence with its global rigid motion. This relation can also be summarized with a unique Jacobian matrix $\mathbf{J}_{\mathrm{E}_k}$, as follows:

$$\mathbf{t}_{\mathrm{E}_k} = \underbrace{\begin{bmatrix} {}^{\mathrm{b}}\mathbf{J}_{\mathrm{E}_k} & {}^k\mathbf{J}_{\mathrm{E}_k} \end{bmatrix}}_{\triangleq \mathbf{J}_{\mathrm{E}_k}} \begin{bmatrix} \mathbf{t}_{\mathrm{b}} \\ \dot{\mathbf{q}}_{\mathrm{a}_k} \end{bmatrix} \tag{6.26}$$

DeNOC relation for tree-type systems: Gathering the previous notations and developments for only one appendage, the relation in (6.24) is written for the whole space robot with all appendages. The notations and definitions introduced in the following are slightly different from the work of S. V. Shah *et al.* in (Shah et al., 2012a), in order to make the derivation of the closed-form equation of dynamics easier. Furthermore, this thesis focuses on a subclass of tree-type systems with the assumption of space robots with a "star" shape, meaning that each appendage is supposed to be a chain-like multi-body system. This hypothesis is used to alleviate the global problem complexity, but the extension to the general case of tree-type space robots would be fairly straightforward using (Shah et al., 2012a; Saha et al., 2013).

Coming back to the kinetic model for one appendage in (6.24), the independent equations for each appendages are stacked together as:

$$\begin{bmatrix} \mathbf{t}_{\mathrm{b}} \\ \mathbf{t}_{\mathrm{a}_1} \\ \mathbf{t}_{\mathrm{a}_2} \\ \vdots \\ \mathbf{t}_{\mathrm{a}_{n_\mathrm{a}}} \end{bmatrix} = \begin{bmatrix} {}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ {}^{1}\mathbf{N}_{\mathrm{b}} & {}^{1}\mathbf{N}_{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ {}^{2}\mathbf{N}_{\mathrm{b}} & \mathbf{0} & {}^{2}\mathbf{N}_{2} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ {}^{n_\mathrm{a}}\mathbf{N}_{\mathrm{b}} & \mathbf{0} & \dots & \mathbf{0} & {}^{n_\mathrm{a}}\mathbf{N}_{n_\mathrm{a}} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_{\mathrm{b}} \\ \dot{\mathbf{q}}_{\mathrm{a}_1} \\ \dot{\mathbf{q}}_{\mathrm{a}_2} \\ \vdots \\ \dot{\mathbf{q}}_{\mathrm{a}_{n_\mathrm{a}}} \end{bmatrix} \tag{6.27}$$

where the matrix ${}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}} = {}^{\mathrm{b}}\mathbf{N}_{d}$ since there is not a twist-propagation matrix between the twist derived from the generalized coordinates in (6.20) and the reference twist for the spacecraft $\mathbf{t}_{\mathrm{b}}$, i.e., ${}^{\mathrm{b}}\mathbf{N}_{\mathrm{b},l} = \mathbf{I}_6$. It would be the case if the generalized coordinates were describing a point of the spacecraft different from the CoM.

To summarize, one advantage of the DeNOC approach is to provide a framework in which the kinematic structure of the space robot is clearly seen from (6.27). Indeed, each appendage is only influenced by the spacecraft and by its joints, but not by any other appendage. This decoupling is only valid at the kinematic level, while the dynamics derived in the sequel brings the unavoidable coupling between all the components of such a complex system. The previous relation can be summarized once again by the classic DeNOC approach by:

$$\mathbf{t} = \underbrace{\mathbf{N}_l \, \mathbf{N}_d}_{\mathbf{N}} \, \dot{\mathbf{q}} \tag{6.28}$$

with the global twist $\mathbf{t} = \begin{bmatrix} \mathbf{t}_{\mathrm{b}}^{\top} & \mathbf{t}_{\mathrm{a}_1}^{\top} & \dots & \mathbf{t}_{\mathrm{a}_{n_\mathrm{a}}}^{\top} \end{bmatrix}^{\top}$, and the generalized coordinates given in (6.11). In this thesis, the decoupled form of $\mathbf{N}$ is defined differently from (Shah et al., 2012a),

as follows:

$$\mathbf{N}_l = \begin{bmatrix} \mathbf{I}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ {}^{1}\mathbf{N}_{\mathrm{b},l} & {}^{1}\mathbf{N}_{1,l} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ {}^{2}\mathbf{N}_{\mathrm{b},l} & \mathbf{0} & {}^{2}\mathbf{N}_{2,l} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ {}^{n_\mathrm{a}}\mathbf{N}_{\mathrm{b},l} & \mathbf{0} & \dots & \mathbf{0} & {}^{n_\mathrm{a}}\mathbf{N}_{n_\mathrm{a},l} \end{bmatrix} \qquad \mathbf{N}_d = \begin{bmatrix} {}^{\mathrm{b}}\mathbf{N}_d & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^{1}\mathbf{N}_d & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & {}^{2}\mathbf{N}_d & \ddots & \vdots \\ \mathbf{0} & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & {}^{n_\mathrm{a}}\mathbf{N}_d \end{bmatrix} \tag{6.29}$$

Is it worth remembering the properties of the sub-blocks related to the appendages in (6.23):

$$\begin{aligned} {}^{k}\mathbf{N}_{k,l} &= \left(\mathbf{I} - {}^{k}\mathbf{A}_k\right)^{-1} \\ {}^{k}\mathbf{N}_{\mathrm{b},l} &= {}^{k}\mathbf{N}_{k,l}\,{}^{k}\mathbf{A}_\mathrm{b} \end{aligned}$$

It allows to state that these properties still holds at the space robot scale due to the lower-triangular shape of $\mathbf{N}_l$. Indeed, inverting it by solving $\mathbf{b} = \mathbf{N}_l\,\mathbf{x}$ for $\mathbf{x}$, an analytic recursive Gaussian elimination yields:

$$\mathbf{N}_l^{-1} = \begin{bmatrix} \mathbf{I}_6 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ -{}^{1}\mathbf{N}_{1,l}^{-1}\,{}^{1}\mathbf{N}_{\mathrm{b},l} & {}^{1}\mathbf{N}_{1,l}^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ -{}^{2}\mathbf{N}_{2,l}^{-1}\,{}^{2}\mathbf{N}_{\mathrm{b},l} & \mathbf{0} & {}^{2}\mathbf{N}_{2,l}^{-1} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ -{}^{n_\mathrm{a}}\mathbf{N}_{n_\mathrm{a},l}^{-1}\,{}^{n_\mathrm{a}}\mathbf{N}_{\mathrm{b},l} & \mathbf{0} & \dots & \mathbf{0} & {}^{n_\mathrm{a}}\mathbf{N}_{n_\mathrm{a},l}^{-1} \end{bmatrix} \tag{6.30}$$

which simplifies with the previous properties to:

$$\mathbf{N}_l = (\mathbf{I} - \mathbf{A})^{-1} \tag{6.31}$$

where the matrix $\mathbf{A}$ is defined by:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ {}^{1}\mathbf{A}_\mathrm{b} & {}^{1}\mathbf{A}_1 & \mathbf{0} & \dots & \mathbf{0} \\ {}^{2}\mathbf{A}_\mathrm{b} & \mathbf{0} & {}^{2}\mathbf{A}_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ {}^{n_\mathrm{a}}\mathbf{A}_\mathrm{b} & \mathbf{0} & \dots & \mathbf{0} & {}^{n_\mathrm{a}}\mathbf{A}_{n_\mathrm{a}} \end{bmatrix} \tag{6.32}$$

6.2.1.3 Constrained Dynamics of the Appendages

Thanks to the recursive twist relation in (6.15), the accelerations also obey the same pattern as (4.36). Differentiating the recursive relation w.r.t. time, one obtains the same general recursion with:

$$\boxed{\dot{\mathbf{t}}_{i_k} = \dot{\mathbf{A}}_{i_k,i_k-1}\,\mathbf{t}_{i_k-1} + \mathbf{A}_{i_k,i_k-1}\,\dot{\mathbf{t}}_{i_k-1} + \dot{\mathbf{P}}_{i_k}\,\dot{\mathbf{q}}_{\mathrm{a}_{i_k}} + \mathbf{P}_{i_k}\,\ddot{\mathbf{q}}_{\mathrm{a}_{i_k}}} \tag{6.33}$$

where the matrices $\left(\mathbf{A}_{i_k,i_k-1}, \dot{\mathbf{A}}_{i_k,i_k-1}\right)$ and $\left(\mathbf{P}_{i_k}, \dot{\mathbf{P}}_{i_k}\right)$ are defined according to the nature of the body considered: (4.30) and (4.38) for a rigid segment, or (5.45) and (5.51) for a flexible one. Regarding the anchorage point of the appendage, the definition of its twist as a function of the spacecraft motion in (6.16) is also differentiated w.r.t. time, as follows:

$$\boxed{\dot{\mathbf{t}}_{0_k} = \dot{\mathbf{A}}_{0_k,\mathrm{b}}\,\mathbf{t}_{\mathrm{b}} + \mathbf{A}_{0_k,\mathrm{b}}\,\dot{\mathbf{t}}_{\mathrm{b}}} \tag{6.34}$$

where the matrix $\dot{\mathbf{A}}_{0_k,\mathrm{b}}$ is defined according to (4.37a) and (4.38a), since the spacecraft is assumed to be rigid:

$$\mathbf{A}_{0_k,\mathrm{b}} = {}^{\mathrm{b}}\boldsymbol{R}_{0_k}^{\top}\,\left(\boldsymbol{\Omega}_{\mathrm{b}}\,\mathbf{A}^{\mathrm{r}}_{0_k,\mathrm{b}} - \mathbf{A}^{\mathrm{r}}_{0_k,\mathrm{b}}\,\boldsymbol{\Omega}_{\mathrm{b}}\right) \tag{6.35}$$

$$= {}^{\mathrm{b}}\boldsymbol{R}_{0_k}^{\top}\,\begin{bmatrix} \mathbf{0}_{3\times3} & -\left(\boldsymbol{\omega}_{\mathrm{b}} \times {}^{\mathrm{b}}\mathbf{r}_{0_k}\right)^{\times} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{6.36}$$

Therefore, the constrained dynamics developed for multi-body systems is still valid for space robots. Generalizing the notations in (4.39) and (5.52) for the rigid and flexible cases, the constrained dynamic equation of any body inside the space robot is denoted by:

$$\boxed{\mathbf{M}_{i_k}\,\dot{\mathbf{t}}_{i_k} + \boldsymbol{\gamma}_{i_k} = \boldsymbol{w}_{i_k} - \mathbf{A}_{i_k+1,i_k}^{\top}\,\boldsymbol{w}_{i_k+1}} \tag{6.37}$$

where $\boldsymbol{\gamma}_i$ denotes the vector of Coriolis and centrifugal terms depending on the body nature:

$$\boldsymbol{\gamma}_{i_k} \Longrightarrow \begin{cases} \boldsymbol{\gamma}_{\mathrm{r}_{i_k}} = \boldsymbol{\Omega}_{i_k}\,\mathbf{M}_{\mathrm{r}_{i_k}}\,\mathbf{E}_v\,\mathbf{t}_{i_k} \\ \boldsymbol{\gamma}_{\mathrm{f}_{i_k}} = \left(\mathring{\mathbf{M}}_{\mathrm{f}_{i_k}} + \boldsymbol{\Omega}_{i_k}\,\mathbf{M}_{\mathrm{f}_{i_k}} + \tilde{\mathbf{M}}_{\mathrm{f}_{i_k}}\right)\,\mathbf{E}_v\,\mathbf{t}_{i_k} + \mathbf{K}_{\mathrm{f}_{i_k}}\,\mathbf{x}_{i_k} \end{cases} \tag{6.38}$$

Finally, the computation of the generalized efforts is obtained with the general relation:

$$\boldsymbol{\tau}_{i_k} = \mathbf{P}_{i_k}^{\top}\,\boldsymbol{w}_{i_k} \tag{6.39}$$

where the size of $(\boldsymbol{w}_{i_k}, \boldsymbol{\tau}_{i_k})$ also depends on the rigid or flexible nature of the body.

6.2.1.4 Constrained Dynamics of the Spacecraft

The spacecraft dynamics is taken from (6.1) and extended by the constraint wrenches coming from each appendage. As mentioned earlier, the kinetic relations are decoupled from one appendage to another, but the dynamics are actually coupling all of them through the spacecraft they are attached to. The dynamic model of this latter reads as follows:

$$\mathbf{M}_\mathrm{b}\dot{\mathbf{t}}_\mathrm{b} + \boldsymbol{\Omega}_\mathrm{b}\,\mathbf{M}_\mathrm{b}\,\mathbf{E}_v\,\mathbf{t}_\mathrm{b} = \boldsymbol{w}_\mathrm{c} + \boldsymbol{w}_\mathrm{e} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{A}_{0_k,\mathrm{b}}^{\top}\boldsymbol{w}_{0_k} \tag{6.40}$$

where the wrench at the anchorage point of an appendage is given by:

$$\boldsymbol{w}_{0_k} = \mathbf{A}_{1_k,0_k}^{\top}\,\boldsymbol{w}_{1_k} \tag{6.41}$$

This wrench is only an intermediate variable to describe the efforts transmitted by the k^{th} appendage to the spacecraft. The wrenches $\boldsymbol{w}_{1_k}$ comes directly from the last step of the dynamic recursion along the k^{th} appendage in (6.37). The use of the transposed of the twist-propagation matrix is a result of screw theory to move the point of expression of any wrench from one point to another. It is similar to (4.19) where the end-tip wrench was expressed at the segment base by multiplying by $\mathbf{A}_{\mathrm{e,o}}^{\top}$. Inspecting the matrix ${}^{k}\mathbf{A}_\mathrm{b}$ in (6.21) and merging the wrenches of each appendage into $\boldsymbol{w}_{\mathrm{a}_k} = \begin{bmatrix} \boldsymbol{w}_{1_k}^{\top} & \dots & \boldsymbol{w}_{n_k}^{\top} \end{bmatrix}^{\top}$, the summation can be re-written as:

$$\sum_{k=1}^{n_\mathrm{a}} \mathbf{A}_{0_k,\mathrm{b}}^{\top}\boldsymbol{w}_{0_k} = \sum_{k=1}^{n_\mathrm{a}} {}^{k}\mathbf{A}_\mathrm{b}^{\top}\,\boldsymbol{w}_{\mathrm{a}_k}$$

Since the wrenches coming from the appendages are internal, all the "external" ones applied on the spacecraft are denoted by $\boldsymbol{w}_\mathrm{b}$, and include the control efforts (see the remark below):

$$\boldsymbol{w}_\mathrm{b} = \mathbf{M}_\mathrm{b}\dot{\mathbf{t}}_\mathrm{b} + \boldsymbol{\Omega}_\mathrm{b}\,\mathbf{M}_\mathrm{b}\,\mathbf{E}_v\,\mathbf{t}_\mathrm{b} + \sum_{k=1}^{n_\mathrm{a}} {}^{k}\mathbf{A}_\mathrm{b}^{\top}\,\boldsymbol{w}_{\mathrm{a}_k} \tag{6.42}$$

Depending on the velocities used to describe the spacecraft, the generalized efforts are obtained by multiplying the wrench by the transposed of their differentiation w.r.t. to the generalized velocities, as shown in Appendix A.3.1. Whatever the model used, equation (6.20) allows to unify the result by giving the generalized efforts as follows, with ${}^{\mathrm{b}}\mathbf{N}_d = \mathbf{P}_\mathrm{b}$:

$$\boldsymbol{\tau}_\mathrm{b} = \mathbf{P}_\mathrm{b}^{\top}\,\boldsymbol{w}_\mathrm{b} \tag{6.43}$$

N.B.: How to account for "internal" control forces?
It is very important to notice that the hypothesis made on the "external" nature of the control efforts implies that the AOCS does not influence the inner dynamics of the space robot. This latter could be the forces and torques produced by the thrusters or by magnetometers. Conversely, if reaction wheels or control moment gyros are considers as actuators, their dynamics strongly influence the spacecraft behavior. In that case, they must be considered as rotating appendages using the previous formalism, and thus integrated in the dynamic model of the space robot itself.

6.2.2 Dynamics Algorithms

The previous constrained equations of motion for each body are assembled to derive the inverse and forward dynamics of a space robot. Similar to the robotic systems, inverse dynamics are obtained in two ways: with the recursive Newton-Euler scheme for simulation, and under the closed-form for control. Eventually, the forward dynamics algorithm of a space robot is derived, based on the general one for tree-type system in (Shah et al., 2012a). The main contribution of this thesis is to provide a straightforward extension to flexible systems, since the algorithm was originally developed for rigid bodies only. Thanks to the flexible models of kinetics in (5.41) and of dynamics in (5.52), the forward dynamics algorithm is extended by merely updating the matrices $\mathbf{A}_{i_k,i_k-1}$, $\mathbf{P}_{i_k}$, $\mathbf{M}_{i_k}$ and their time derivatives with their flexible counterparts.

6.2.2.1 Inverse dynamics

The inverse dynamics scheme consists in obtaining the generalized efforts of the spacecraft and of each appendage. The classic Newton-Euler is extended to space robot using (Carignan and Akin, 2000). The closed-form equation of dynamics are then derived with the DeNOC approach to obtain the general model presented in (Nenchev, 2013).

In this latter, the dynamic model of a space robot is partitioned as follows:

$$\begin{bmatrix} \mathbf{D}_{\mathrm{bb}} & \mathbf{D}_{\mathrm{ba}} \\ \mathbf{D}_{\mathrm{ba}}^{\top} & \mathbf{D}_{\mathrm{aa}} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_{\mathrm{b}} \\ \ddot{\mathbf{q}}_{\mathrm{a}} \end{bmatrix} + \begin{bmatrix} \mathbf{h}_{\mathrm{b}}(\mathbf{q},\dot{\mathbf{q}}) \\ \mathbf{h}_{\mathrm{a}}(\mathbf{q},\dot{\mathbf{q}}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_{\mathrm{b}} \\ \boldsymbol{\tau}_{\mathrm{a}} \end{bmatrix} + \begin{bmatrix} {}^{\mathrm{b}}\mathbf{J}_{\mathrm{E_a}}^{\top} \\ {}^{\mathrm{a}}\mathbf{J}_{\mathrm{E_a}}^{\top} \end{bmatrix} \boldsymbol{w}_{\mathrm{E_a}} \tag{6.44}$$

with

- $(\mathbf{D}_{\mathrm{bb}}, \mathbf{D}_{\mathrm{ba}}, \mathbf{D}_{\mathrm{aa}})$ partitioning of the global mass matrix
- $(\mathbf{h}_{\mathrm{b}}, \mathbf{h}_{\mathrm{a}})$ partitioning of the Coriolis and centrifugal terms
- $\boldsymbol{\tau}_{\mathrm{b}}$ generalized efforts of the spacecraft
- $\boldsymbol{\tau}_{\mathrm{a}}$ generalized efforts of the appendages
- $\boldsymbol{w}_{\mathrm{E_a}}$ external efforts applied at the end-effectors
- $\left({}^{\mathrm{b}}\mathbf{J}_{\mathrm{E_a}}, {}^{\mathrm{a}}\mathbf{J}_{\mathrm{E_a}}\right)$ partitioning of the Jacobian matrix of the end-effectors

In the sequel, the general model for a multi-appendages space robot is developed to describe the case of star-shaped multi-body systems with flexible bodies.

Newton-Euler recursive scheme Inverse dynamics are first derived for simulation purpose by extending the recursive Newton-Euler scheme introduced earlier. Provided that the spacecraft twist and its time derivative are known, a first kinetic loop computes the twist and the acceleration of the anchorage point through (6.16) and (6.34), and then iterates over the sub-bodies of the appendage with (6.15) and (6.33):

$$\left(\mathbf{t}_{\mathrm{b}}, \dot{\mathbf{t}}_{\mathrm{b}}\right) \underset{\forall k=1\ldots n_{\mathrm{a}}}{\Longrightarrow} \left(\mathbf{t}_{0_k}, \dot{\mathbf{t}}_{0_k}\right) \underset{\forall i_k=1_k\ldots n_k}{\Longrightarrow} \left(\mathbf{t}_{i_k}, \dot{\mathbf{t}}_{i_k}\right)$$

Then, a second dynamic loop starts from the end-effector of each appendage, iterates along each sub-bodies to compute their wrenches with (6.37), derives the generalized effort by projecting them along the motion axes with (6.39), and eventually provides the wrench applied on the spacecraft by $\boldsymbol{w}_{0_k}$ with (6.41). Summing these efforts over all the appendages, the generalized effort of the spacecraft are obtained by (6.42) and (6.43).

$$\underset{\forall k=1\ldots n_{\mathrm{a}}}{\Longrightarrow} (\boldsymbol{w}_{n_k+1}) \underset{\forall i_k=n_k\ldots 1_k}{\Longrightarrow} (\boldsymbol{w}_{i_k}, \boldsymbol{\tau}_{i_k}) \Longrightarrow (\boldsymbol{w}_{0_k}) \underset{\sum_k}{\Longrightarrow} (\boldsymbol{\tau}_{\mathrm{b}})$$

The global algorithm gathering these steps is summarized in Algorithm 6.1. The superscript {r/f} is used to emphasize the matrices to update with the rigid or flexible models. The whole algorithm is written to stay valid for both types of bodies.

Closed-form dynamics The closed-form equation of dynamics is now derived for control purpose and system analysis. It is obtained by gathering the dynamic equations of the spacecraft, and of the appendages. With the previous results for one appendage in (6.37) and (6.38), the set of equations is similar to (4.51) and (5.61) and reads as follows in the general flexible case with the current notations:

$$[\mathbf{M}_{\mathrm{a}_k}]\ \dot{\mathbf{t}}_{\mathrm{a}_k} + \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]\ \mathbf{t}_{\mathrm{a}_k} + \left[\mathbf{K}_{\mathrm{f}_{\mathrm{a}_k}}\right]\ \mathbf{x}_{\mathrm{a}_k} = (\mathbf{I} - {}^{k}\mathbf{A}_k)^{\top}\ \boldsymbol{w}_{\mathrm{a}_k} + \overline{\mathbf{A}}^{\top}_{n_k+1,n_k}\ \boldsymbol{w}_{\mathrm{E}_k} \tag{6.45}$$

Algorithm 6.1: Newton-Euler algorithm for space robot inverse dynamics (`InvDynSpace`)

Function : $\boldsymbol{\tau} = \texttt{InvDynSpace}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$

Input : $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$

Output : $\boldsymbol{\tau}$

Data : $\left\{ {}^{i_k-1}R^{\mathrm{r/f}}_{i_k}\,,\ \mathbf{A}^{\mathrm{r/f}}_{i_k,i_k-1}\,,\ \mathbf{P}_{i_k}\,,\ \mathbf{M}_{i_k}\,,\ \mathbf{A}_{0_k,\mathrm{b}}\,,\ \mathbf{M}_{\mathrm{b}}\,,\ \mathbf{P}_{\mathrm{b}}\ ;\ i_k = 0_k \ldots n_k + 1\,,\ k = 1 \ldots n_{\mathrm{a}} \right\}$

`Initialize` $\left(\boldsymbol{w}_{\mathrm{a/b}} = \mathbf{0}_{6\times 1}\right)$ // **APPENDAGE DYNAMICS**

$\mathbf{t}_{\mathrm{b}} = \mathbf{P}_{\mathrm{b}}\,\dot{\mathbf{q}}_{\mathrm{b}}$ (6.20)

$\dot{\mathbf{t}}_{\mathrm{b}} = \mathbf{P}_{\mathrm{b}}\,\ddot{\mathbf{q}}_{\mathrm{b}} + \boldsymbol{\Omega}_{\mathrm{b}}\,\mathbf{P}_{\mathrm{b}}\,\dot{\mathbf{q}}_{\mathrm{b}}$ (6.7)

$\dot{\mathbf{A}}_{0_k,\mathrm{b}} = {}^{0_k}\boldsymbol{R}^{\top}_{\mathrm{b}}\left(\boldsymbol{\Omega}_{\mathrm{b}}\mathbf{A}^{\mathrm{r}}_{0_k,\mathrm{b}} - \mathbf{A}^{\mathrm{r}}_{0_k,\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}}\right)$ (6.17) and (6.18)

for $k = 1 \ldots n_{\mathrm{a}}$ **do**

  `Initialize` $\left(\mathbf{t}_{0_k} = \mathbf{A}_{0_k,\mathrm{b}}\mathbf{t}_{\mathrm{b}},\ \dot{\mathbf{t}}_{0_k} = \mathbf{A}_{0_k,\mathrm{b}}\dot{\mathbf{t}}_{\mathrm{b}} + \dot{\mathbf{A}}_{0_k,\mathrm{b}}\mathbf{t}_{\mathrm{b}},\ \mathbf{A}^{\mathrm{r}}_{1_k,0_k},\ \dot{\mathbf{A}}^{\mathrm{r}}_{1_k,0_k} = \mathbf{0}_{m_{f_{1_k}}\times 6}\right)$ (6.16)

  for $i = 1_k \ldots n_k + 1$ **do** // Recursive computation and storage of $\mathbf{A}_{i_k,i_k-1}$, $\dot{\mathbf{A}}_{i_k,i_k-1}$, and $\boldsymbol{\Omega}_{i_k}$

    ${}^{i_k}R_{i_k-1} = {}^{i_k-1}R^{\mathrm{r}}_{i_k}(q_{i_k})^{\top}$ or ${}^{i_k-1}R^{\mathrm{f}}_{i_k}(q_{i_k}, \boldsymbol{\delta}_{i_k-1})^{\top}$ (4.22) and (5.35)

    ${}^{i_k}\boldsymbol{R}_{i_k-1} = \mathrm{diag}({}^{i_k}R_{i_k-1}, {}^{i_k}R_{i_k-1})$ (4.31)

    $\mathbf{A}_{i_k,i_k-1} = {}^{i_k}\boldsymbol{R}_{i_k-1}\,\mathbf{A}^{\mathrm{r/f}}_{i_k,i_k-1}$ (4.30a) and (5.45a)

    $\dot{\mathbf{A}}_{i_k,i_k-1} = {}^{i_k}\boldsymbol{R}_{i_k-1}\,\dot{\mathbf{A}}^{\mathrm{r/f}}_{i_k,i_k-1}$ (4.38a) and (5.51a)

    $\mathbf{t}_{i_k} = \mathbf{A}_{i_k,i_k-1}\,\mathbf{t}_{i_k-1} + \mathbf{P}_{i_k}\,\boldsymbol{q}_{i_k}$ (4.28)

    $\dot{\mathbf{t}}_{i_k} = \mathbf{A}_{i_k,i_k-1}\,\dot{\mathbf{t}}_{i_k-1} + \dot{\mathbf{A}}_{i_k,i_k-1}\,\mathbf{t}_{i_k-1} + \mathbf{P}_{i_k}\,\ddot{\boldsymbol{q}}_{i_k} + \boldsymbol{\Omega}_{i_k}\,\mathbf{P}_{i_k}\,\dot{\boldsymbol{q}}_{i_k}$ (4.36)

    if $i_k \leq n_k$ **then**

      $\dot{\mathbf{A}}^{\mathrm{r/f}}_{i_k+1,i_k} = \mathring{\mathbf{A}}^{\mathrm{f}}_{i_k+1,i_k} + \boldsymbol{\Omega}_{i_k}\,\mathbf{A}^{\mathrm{r/f}}_{i_k+1,i_k} - \mathbf{A}^{\mathrm{r/f}}_{i_k+1,i_k}\,\bar{\boldsymbol{\Omega}}_{i_k}$ (4.37a) and (5.49)

    end

  end

  `Initialize` $\left(\boldsymbol{w}_{n_k+1} = \mathbf{M}_{n_k+1}\,\dot{\mathbf{t}}_{n_k+1} + \boldsymbol{\gamma}_{n_k+1}\right)$

  for $i_k = n_k \ldots 1_k$ **do** // Recursive computation of $\widehat{\mathbf{M}}_{i_k}, \dot{\widehat{\mathbf{M}}}_{i_k}, \widehat{\mathbf{H}}_{i_k}$

    $\boldsymbol{\gamma}_{i_k} = \left(\dot{\mathbf{M}}_{\mathrm{r}_{i_k}}\,\mathbf{t}_{i_k}\right)$ or $\left(\dot{\mathbf{M}}_{\mathrm{f}_{i_k}}\,\mathbf{t}_{i_k} + \mathbf{K}_{\mathrm{f}_{i_k}}\,\mathbf{x}_{i_k}\right)$ (6.38)

    $\boldsymbol{W}_{i_k} = \mathbf{M}_{i_k}\dot{\mathbf{t}}_{i_k} + \boldsymbol{\gamma}_{i_k}$ (4.39) and (5.52)

    $\boldsymbol{w}_{i_k} = \boldsymbol{W}_{i_k} + \mathbf{A}^{\top}_{i_k+1,i_k}\boldsymbol{w}_{i_k+1}$

    $\boldsymbol{\tau}_{i_k} = \mathbf{P}^{\top}_{i_k}\boldsymbol{w}_{i_k}$ (4.42) and (5.54)

  end

  $\boldsymbol{w}_{\mathrm{a/b}} = \boldsymbol{w}_{\mathrm{a/b}} - \left(\mathbf{A}_{1_k,0_k}\mathbf{A}_{0_k,\mathrm{b}}\right)^{\top}\boldsymbol{w}_{1_k}$

end

// **SPACECRAFT DYNAMICS**

$\boldsymbol{w}_{\mathrm{b}} = \mathbf{M}_{\mathrm{b}}\dot{\mathbf{t}}_{\mathrm{b}} + \boldsymbol{\Omega}_{\mathrm{b}}\,\mathbf{M}_{\mathrm{b}}\,\mathbf{E}_v\,\mathbf{t}_{\mathrm{b}} - \boldsymbol{w}_{\mathrm{a/b}}$ (6.42)

$\boldsymbol{\tau}_{\mathrm{b}} = \mathbf{P}^{\top}_{\mathrm{b}}\,\boldsymbol{w}_{\mathrm{b}}$ (6.43)

return $\boldsymbol{\tau}$

where the abuse of notation for $\dot{\mathbf{M}}$ is introduced in (A.25) of Appendix A.3.3, and is valid for both rigid and flexible cases. Only the stiffness term would disappear if a fully rigid appendage were considered.

When the spacecraft model in (6.37) is merged with the previous model of each appendage, the global set of dynamic equations yields:

$$\left\{\begin{array}{lllllll} \mathbf{M}_{\mathrm{b}}\dot{\mathbf{t}}_{\mathrm{b}} & + & \dot{\mathbf{M}}_{\mathrm{b}}\mathbf{t}_{\mathrm{b}} & = & \boldsymbol{w}_{\mathrm{b}} & - & \sum_{k=1}^{n_{\mathrm{a}}} {}^{k}\mathbf{A}_{\mathrm{b}}^{\top}\,\boldsymbol{w}_{\mathrm{a}_k} \\ \hline \left[\mathbf{M}_{\mathrm{a}_1}\right]\dot{\mathbf{t}}_{\mathrm{a}_1} & + & \left[\dot{\mathbf{M}}_{\mathrm{a}_1}\right]\mathbf{t}_{\mathrm{a}_1} + \left[\mathbf{K}_{\mathrm{f}_{\mathrm{a}_1}}\right]\mathbf{x}_{\mathrm{a}_1} & = & (\mathbf{I} - {}^{1}\mathbf{A}_1)^{\top}\boldsymbol{w}_{\mathrm{a}_1} & + & \overline{\mathbf{A}}^{\top}_{n_1+1,n_1}\boldsymbol{w}_{\mathrm{E}_1} \\ & & & \vdots & & & \\ \left[\mathbf{M}_{\mathrm{a}_{n_{\mathrm{a}}}}\right]\dot{\mathbf{t}}_{\mathrm{a}_{n_{\mathrm{a}}}} & + & \left[\dot{\mathbf{M}}_{\mathrm{a}_{n_{\mathrm{a}}}}\right]\mathbf{t}_{\mathrm{a}_{n_{\mathrm{a}}}} + \left[\mathbf{K}_{\mathrm{f}_{\mathrm{a}_{n_{\mathrm{a}}}}}\right]\mathbf{x}_{\mathrm{a}_{n_{\mathrm{a}}}} & = & (\mathbf{I} - {}^{n_{\mathrm{a}}}\mathbf{A}_{n_{\mathrm{a}}})^{\top}\boldsymbol{w}_{\mathrm{a}_{n_{\mathrm{a}}}} & + & \overline{\mathbf{A}}^{\top}_{n_{n_{\mathrm{a}}}+1,n_{n_{\mathrm{a}}}}\boldsymbol{w}_{\mathrm{E}_{n_{\mathrm{a}}}} \end{array}\right.$$

Reminding the notations used to denote the global DeNOC matrix $\mathbf{N}_l$ in (6.29) and its inverse in (6.30), the previous system can be summarized as:

$$[\mathbf{M}]\,\dot{\mathbf{t}} + \left[\dot{\mathbf{M}}\right]\,\mathbf{t} + [\mathbf{K}_{\mathrm{f}}]\,\mathbf{x} = \mathbf{N}_l^{-\top}\,\boldsymbol{w} + \left[\overline{\mathbf{A}}_{\mathrm{E}}^{\top}\right]\,\boldsymbol{w}_{\mathrm{E}} \tag{6.46}$$

using the diagonal matrix notation introduced earlier, and the global vectors defined as follows:

$$\begin{array}{rcl} \mathbf{t} & = & \left[\begin{array}{cccc} \mathbf{t}_{\mathrm{b}}^{\top} & \mathbf{t}_{\mathrm{a}_1}^{\top} & \ldots & \mathbf{t}_{\mathrm{a}_{n_{\mathrm{a}}}}^{\top} \end{array}\right]^{\top} \\ \boldsymbol{w} & = & \left[\begin{array}{cccc} \boldsymbol{w}_{\mathrm{b}}^{\top} & \boldsymbol{w}_{\mathrm{a}_1}^{\top} & \ldots & \boldsymbol{w}_{\mathrm{a}_{n_{\mathrm{a}}}}^{\top} \end{array}\right]^{\top} \\ \boldsymbol{w}_{\mathrm{E}} & = & \left[\begin{array}{ccc} \boldsymbol{w}_{\mathrm{E}_1}^{\top} & \ldots & \boldsymbol{w}_{\mathrm{E}_{n_{\mathrm{a}}}}^{\top} \end{array}\right]^{\top} \\ [\mathbf{M}] & = & \mathrm{diag}\left(\mathbf{M}_{\mathrm{b}},\, \{[\mathbf{M}_{\mathrm{a}_k}],\, k = 1 \ldots n_{\mathrm{a}}\}\right) \\ \left[\dot{\mathbf{M}}\right] & = & \mathrm{diag}\left(\dot{\mathbf{M}}_{\mathrm{b}},\, \left\{\left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right],\, k = 1 \ldots n_{\mathrm{a}}\right\}\right) \\ [\mathbf{K}_{\mathrm{f}}] & = & \mathrm{diag}\left(\mathbf{0}_{6\times 6},\, \left\{\left[\mathbf{K}_{\mathrm{f}_{\mathrm{a}_k}}\right],\, k = 1 \ldots n_{\mathrm{a}}\right\}\right) \\ \left[\overline{\mathbf{A}}_{\mathrm{E}}^{\top}\right] & = & \left[\begin{array}{c} \mathbf{0}_{6\times 6n_{\mathrm{a}}} \\ \mathrm{diag}(\overline{\mathbf{A}}^{\top}_{n_k+1,n_k},\, k = 1 \ldots n_{\mathrm{a}}) \end{array}\right] \end{array}$$

Applying again the Kane's formalism by pre-multiplying with the global DeNOC matrix in (6.28) and developing the twist accordingly, the closed-form equation of dynamics is obtained for the whole space robot as:

$$\mathbf{N}^{\top}\,[\mathbf{M}]\,\mathbf{N}\,\ddot{\mathbf{q}} + \mathbf{N}^{\top}\,[\mathbf{M}]\,\dot{\mathbf{N}}\,\dot{\mathbf{q}} + \mathbf{N}^{\top}\,\left[\dot{\mathbf{M}}\right]\,\mathbf{N}\,\dot{\mathbf{q}} + \mathbf{N}^{\top}\,[\mathbf{K}_{\mathrm{f}}]\,\mathbf{x} = \mathbf{N}^{\top}\,\mathbf{N}_l^{-\top}\,\boldsymbol{w} + \mathbf{N}^{\top}\,\left[\overline{\mathbf{A}}_{\mathrm{E}}^{\top}\right]\,\boldsymbol{w}_{\mathrm{E}}$$

The stiffness terms can be developed in as similar way as for the flexible dynamics in (5.62), where it was shown that:

$$\begin{aligned} {}^{k}\mathbf{N}_{k}^{\top}\ \left[\mathbf{K}_{\mathrm{f}_{\mathrm{a}_k}}\right]\mathbf{x}_{\mathrm{a}_k} &= {}^{k}\mathbf{N}_{d}^{\top}\,{}^{k}\mathbf{N}_{k,l}^{\top}\ \left[\mathbf{K}_{\mathrm{f}_{\mathrm{a}_k}}\right]\mathbf{x}_{\mathrm{a}_k} \\ &= {}^{k}\mathbf{N}_{d}^{\top}\ \left[\mathbf{K}_{\mathrm{f}_{\mathrm{a}_k}}\right]\mathbf{x}_{\mathrm{a}_k} \\ &= \mathbf{K}_{\mathrm{a}_k}\,\mathbf{q}_{\mathrm{a}_k} \end{aligned}$$

Developing $\mathbf{N}^{\top}\ [\mathbf{K}_{\mathrm{f}}]\ \mathbf{x}$, one obtains that all the appendages rows satisfies this same relation, but the first row related to the spacecraft yields:

$$\begin{aligned} \sum_{k}{}^{k}\mathbf{N}_{b}^{\top}\ \left[\mathbf{K}_{\mathrm{f}_{\mathrm{a}_k}}\right]\mathbf{x}_{\mathrm{a}_k} &= \sum_{k}\mathbf{P}_{\mathrm{b}}^{\top\,k}\mathbf{A}_{\mathrm{b}}^{\top}\,{}^{k}\mathbf{N}_{k,l}^{\top}\ \left[\mathbf{K}_{\mathrm{f}_{\mathrm{a}_k}}\right]\mathbf{x}_{\mathrm{a}_k} \\ &= \sum_{k}\mathbf{P}_{\mathrm{b}}^{\top\,k}\mathbf{A}_{\mathrm{b}}^{\top}\ \left[\mathbf{K}_{\mathrm{f}_{\mathrm{a}_k}}\right]\mathbf{x}_{\mathrm{a}_k} \\ &= \sum_{k}\mathbf{P}_{\mathrm{b}}^{\top}\mathbf{A}_{1_k,\mathrm{b}}^{\top}\,\mathbf{K}_{\mathrm{f}_{1_k}}\,\mathbf{x}_{1_k} \\ &= \mathbf{0} \end{aligned}$$

since the same argument of the flexible dynamics holds, that $\mathbf{A}_{i_k,i_k-1}^{\top}\,\mathbf{K}_{\mathrm{f}_{i_k}} = \mathbf{0}$. As a conclusion, the stiffness terms can be re-written as a function of the bodies stiffness matrices as:

$$\mathbf{N}^{\top}\ [\mathbf{K}_{\mathrm{f}}]\ \mathbf{x} = \mathrm{diag}\left(\mathbf{0}_{6\times 6},\ \{\mathbf{K}_{\mathrm{a}_k},\ k = 1\ldots n_{\mathrm{a}}\}\right)\ \mathbf{q}$$

with

$$\mathbf{K}_{\mathrm{a}_k} = \mathrm{diag}(\begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{\delta_{i_k}\delta_{i_k}} \end{bmatrix},\ i_k = 1_k\ldots n_k)$$

Regarding the RHS, $\mathbf{N}^{\top}\,\mathbf{N}_{l}^{-\top}\,\boldsymbol{w}$ reduces to the generalized efforts, as expected:

$$\mathbf{N}_{d}^{\top}\boldsymbol{w} = \begin{bmatrix} \boldsymbol{\tau}_{\mathrm{b}}^{\top} & \boldsymbol{\tau}_{\mathrm{a}_1}^{\top} & \ldots & \boldsymbol{\tau}_{\mathrm{a}_{n_{\mathrm{a}}}}^{\top} \end{bmatrix}^{\top} \triangleq \boldsymbol{\tau}$$

Furthermore, the computation of $\mathbf{N}^{\top}\left[\overline{\mathbf{A}}_{\mathrm{E}}^{\top}\right]$ brings directly the transposed of the global Ja-

cobian matrix of the end-effectors, gathering the ones given in (6.25), as follows:

$$\mathbf{N}^\top \left[\overline{\mathbf{A}}_{\mathrm{E}}^\top\right] \boldsymbol{w}_{\mathrm{E}} = \underbrace{\begin{bmatrix} {}^{\mathrm{b}}\mathbf{J}_{\mathrm{E}_1}^\top & {}^{\mathrm{b}}\mathbf{J}_{\mathrm{E}_2}^\top & \dots & {}^{\mathrm{b}}\mathbf{J}_{\mathrm{E}_{n_\mathrm{a}}}^\top \\ {}^{1}\mathbf{J}_{\mathrm{E}_1}^\top & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & {}^{2}\mathbf{J}_{\mathrm{E}_2}^\top & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & {}^{\mathrm{b}}\mathbf{J}_{\mathrm{E}_{n_\mathrm{a}}}^\top \end{bmatrix}}_{\triangleq \mathbf{J}_{\mathrm{E}}^\top} \boldsymbol{w}_{\mathrm{E}}$$

Dynamics of a space robot

To summarize the previous computations, the space robot dynamics is reduced to the minimum set of equations governing the behavior of its spacecraft and of each appendage by:

$$\mathbf{D}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{K}\,\mathbf{q} = \boldsymbol{\tau} + \mathbf{J}_{\mathrm{E}}^\top\,\boldsymbol{w}_{\mathrm{E}} \tag{6.47}$$

with

$$\mathbf{D}(\mathbf{q}) = \mathbf{N}^\top\,[\mathbf{M}]\,\mathbf{N} \tag{6.48}$$

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{N}^\top\,[\mathbf{M}]\,\dot{\mathbf{N}} + \mathbf{N}^\top\left[\dot{\mathbf{M}}\right]\,\mathbf{N} \tag{6.49}$$

$$\mathbf{K} = \mathrm{diag}\left(\mathbf{0}_{6\times 6},\,\{\mathbf{K}_{\mathrm{a}_k},\,k = 1\dots n_\mathrm{a}\}\right) \tag{6.50}$$

$$\boldsymbol{\tau} = \mathbf{N}_d^\top\,\boldsymbol{w} \tag{6.51}$$

When the mass matrix is developed according to the structure of $\mathbf{N}$ given in (6.27), one obtains the following sparse structure that clearly emphasize the dynamic coupling between the appendages and the spacecraft:

$$\mathbf{D} = \left[\begin{array}{c|cccc} \mathbf{D}_{\mathrm{bb}} & \mathbf{D}_{\mathrm{ba}_1} & \dots & \dots & \mathbf{D}_{\mathrm{ba}_{n_\mathrm{a}}} \\ \hline * & \mathbf{D}_{\mathrm{a}_1} & \mathbf{0} & \dots & \mathbf{0} \\ * & \mathbf{0} & \ddots & \ddots & \vdots \\ * & \vdots & \ddots & \ddots & \mathbf{0} \\ * & \mathbf{0} & \dots & \mathbf{0} & \mathbf{D}_{\mathrm{a}_{n_\mathrm{a}}} \end{array}\right] \tag{6.52}$$

where the mass matrices appearing on the diagonal are the fixed-base models introduced earlier in this thesis. Each sub-block of the global mass matrix is developed as follows, for the spacecraft, the coupling terms, and the appendages:

$$\mathbf{D}_{\mathrm{bb}} = {}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}}^{\top}\mathbf{M}_{\mathrm{b}}{}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}} + \textstyle\sum_k {}^{k}\mathbf{N}_{\mathrm{b}}^{\top}\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^{k}\mathbf{N}_{\mathrm{b}} \tag{6.53a}$$

$$\mathbf{D}_{\mathrm{ba}_k} = {}^{k}\mathbf{N}_{\mathrm{b}}^{\top}\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^{k}\mathbf{N}_{k} \tag{6.53b}$$

$$\mathbf{D}_{\mathrm{a}_k} = {}^{k}\mathbf{N}_{k}^{\top}\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^{k}\mathbf{N}_{k} \tag{6.53c}$$

In the same fashion, an interesting insight is given into the structure of the nonlinear efforts by computing the matrix $\mathbf{C}$:

$$\begin{aligned}\mathbf{C} &= \mathbf{N}^{\top}\left[\mathbf{M}\right]\dot{\mathbf{N}} + \mathbf{N}^{\top}\left[\dot{\mathbf{M}}\right]\mathbf{N} \\ &= \left[\begin{array}{c|cccc} \mathbf{C}_{\mathrm{bb}} & \mathbf{C}_{\mathrm{ba}_1} & \dots & \dots & \mathbf{C}_{\mathrm{ba}_{n_\mathrm{a}}} \\ \hline \mathbf{C}_{\mathrm{a}_1\mathrm{b}} & \mathbf{C}_{\mathrm{a}_1} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \mathbf{0} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{C}_{\mathrm{a}_{n_\mathrm{a}}\mathrm{b}} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{C}_{\mathrm{a}_{n_\mathrm{a}}} \end{array}\right]\end{aligned} \tag{6.54}$$

where again the sub-matrices $\mathbf{C}_{\mathrm{a}_k}$ are obtained for fixed-base appendages, while the central and coupling terms read:

$$\mathbf{C}_{\mathrm{bb}} = {}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}}^{\top}\left(\mathbf{M}_{\mathrm{b}}{}^{\mathrm{b}}\dot{\mathbf{N}}_{\mathrm{b}} + \dot{\mathbf{M}}_{\mathrm{b}}{}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}}\right) + \textstyle\sum_k {}^{k}\mathbf{N}_{\mathrm{b}}^{\top}\left(\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^{k}\dot{\mathbf{N}}_{\mathrm{b}} + \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^{k}\mathbf{N}_{\mathrm{b}}\right) \tag{6.55a}$$

$$\mathbf{C}_{\mathrm{ba}_k} = {}^{k}\mathbf{N}_{\mathrm{b}}^{\top}\left(\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^{k}\dot{\mathbf{N}}_{k} + \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^{k}\mathbf{N}_{k}\right) \tag{6.55b}$$

$$\mathbf{C}_{\mathrm{a}_k\mathrm{b}} = {}^{k}\mathbf{N}_{k}^{\top}\left(\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^{k}\dot{\mathbf{N}}_{\mathrm{b}} + \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^{k}\mathbf{N}_{\mathrm{b}}\right) \tag{6.55c}$$

$$\mathbf{C}_{\mathrm{a}_k} = {}^{k}\mathbf{N}_{k}^{\top}\left(\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^{k}\dot{\mathbf{N}}_{k} + \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^{k}\mathbf{N}_{k}\right) \tag{6.55d}$$

which leads to the following decoupling of the Coriolis and centrifugal vectors for each appendage:

$$\mathbf{h}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} = \left[\begin{array}{c} \mathbf{h}_{\mathrm{b}}\left(\dot{\mathbf{q}}_{\mathrm{b}},\,\mathbf{q}_{\mathrm{a}},\,\dot{\mathbf{q}}_{\mathrm{a}}\right) \\ \hline \mathbf{h}_{\mathrm{a}_1}\left(\dot{\mathbf{q}}_{\mathrm{b}},\,\mathbf{q}_{\mathrm{a}_1},\,\dot{\mathbf{q}}_{\mathrm{a}_1}\right) \\ \vdots \\ \mathbf{h}_{\mathrm{a}_{n_\mathrm{a}}}\left(\dot{\mathbf{q}}_{\mathrm{b}},\,\mathbf{q}_{\mathrm{a}_{n_\mathrm{a}}},\,\dot{\mathbf{q}}_{\mathrm{a}_{n_\mathrm{a}}}\right) \end{array}\right] \tag{6.56}$$

The algorithms from the robotic modeling in appendix A.3 are extended to the space robot in appendix C.1 and allow to compute recursively the mass matrix $\mathbf{D}$ with Algorithm C.1, and the Coriolis and centrifugal matrix $\mathbf{C}$ or the corresponding vector $\mathbf{h}$ with Algorithm C.2.

6.2.2.2 Forward dynamics

The forward dynamics algorithm based on the DeNOC approach is extended to space robot in Algorithm C.3 by adapting the more general algorithm for tree-type systems in (Shah et al., 2012a) to the case of "star-shaped" space robots.

The main idea of this algorithm is to write the DeNOC approach for tree-type systems in the same way as the chain-like one. To this end, the whole system is split into modules, each of them being a chain-like system. Then the twist propagation matrices are written between the modules themselves instead of the segments, and a global matrix $\mathbf{N} = \mathbf{N}_l\,\mathbf{N}_d$ is derived based on the inter-module connexions. The decoupled form of this matrix can be written as follows with the current notations:

$$\mathbf{N} = \underbrace{\left[\begin{array}{c|cccc} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline {}^{1}\mathbf{N}_{\mathrm{b},l} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ {}^{2}\mathbf{N}_{\mathrm{b},l} & \mathbf{0} & \mathbf{I} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ {}^{n_\mathrm{a}}\mathbf{N}_{\mathrm{b},l} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{I} \end{array}\right]}_{=\mathbf{N}_l} \underbrace{\left[\begin{array}{ccccc} {}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^{1}\mathbf{N}_{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & {}^{2}\mathbf{N}_{2} & \ddots & \vdots \\ \mathbf{0} & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & {}^{n_\mathrm{a}}\mathbf{N}_{n_\mathrm{a}} \end{array}\right]}_{=\mathbf{N}_d} \tag{6.57}$$

This decomposition is slightly different from the one introduced in (6.29). This new form allows to perform exactly the same recursions presented for the chain-like forward dynamics in Algorithm A.3, since $\mathbf{N}_l$ has identity matrices on the diagonal. The three step algorithm

of forward dynamics can thus be used with the following matrices defined for each module:

$$\begin{aligned}
\mathbf{P}_i &\Longrightarrow {}^{i}\mathbf{N}_i \\
\mathbf{A}_{i,j} &\Longrightarrow \mathbf{0}, \quad \forall i \geq 1, \forall j < i \\
\mathbf{A}_{i,\mathrm{b}} &\Longrightarrow {}^{i}\mathbf{N}_{\mathrm{b},l} \\
\mathbf{M}_i &\Longrightarrow [\mathbf{M}_{\mathrm{a}_i}]
\end{aligned}$$

The new twist-propagation matrices represent the connexion between different modules. Under the hypothesis of a "star-shaped" space robot, there is not any direct link between two appendages, and as a consequence, the lower diagonal part of $\mathbf{N}_l$ is zero, except for the spacecraft column. Indeed, these terms represent the anchorage of each appendage on a specific point of the spacecraft.

The resulting forward dynamics algorithm is not directly written with the equivalent matrices for each module, because it would lead to invert the inertia matrix of each module at one point (i.e., when the diagonal terms $\mathbf{\Delta}_i$ are inverted at the second step of Algorithm A.3). Instead, an inward recursive algorithm is derived by starting at each child module separately, and then by switching to its parent. This approach is presented in (Shah et al., 2012a) for the general case of a tree-type system and applied on humanoid robots. A reduced version is used in this thesis to focus on one parent module, i.e., the spacecraft, with multiple child modules attached to it. The resulting algorithm is given in Algorithm C.3 of Appendix C.1.

6.2.3 Simplifying hypothesis about orbital dynamics

A last simplifying hypothesis is made on the dynamics of the space robot to decouple the orbital mechanics from the multi-body dynamics. Considering that a spacecraft is following a given orbit, its linear velocity under the terrestrial gravity satisfies the dynamics in (6.1):

$$m_{\mathrm{b}}\, \dot{\boldsymbol{v}}_{\mathrm{b}} = \boldsymbol{f}_{\mathrm{g}} + \boldsymbol{f}_{\mathrm{dist}} + \boldsymbol{f}_{\mathrm{c}}$$

where $\left(\boldsymbol{f}_{\mathrm{g}},\, \boldsymbol{f}_{\mathrm{dist}},\, \boldsymbol{f}_{\mathrm{c}}\right)$ denote respectively, the gravity force, the disturbances, and the control forces. It is recalled that $\boldsymbol{f}_{\mathrm{g}} = \frac{\mu\, m_{\mathrm{b}}}{r_{\mathrm{b}}^2} \boldsymbol{u}_{\mathrm{b}}$.

Without any external effort nor control, the gravitational speed $\boldsymbol{v}_{\mathrm{b_g}}$ would satisfies:

$$m_{\mathrm{b}}\, \dot{\boldsymbol{v}}_{\mathrm{b_g}} = \boldsymbol{f}_{\mathrm{g}}$$

Splitting the general speed with a perturbation term as $\boldsymbol{v}_{\mathrm{b}} = \boldsymbol{v}_{\mathrm{b_g}} + \delta\boldsymbol{v}_{\mathrm{b}}$, the full model is re-written as follows:

$$m_{\mathrm{b}} \left(\dot{\boldsymbol{v}}_{\mathrm{b_g}} + \delta\dot{\boldsymbol{v}}_{\mathrm{b}} \right) = \boldsymbol{f}_{\mathrm{g}} + \boldsymbol{f}_{\mathrm{dist}} + \boldsymbol{f}_{\mathrm{c}}$$

For a short amount of time, the spacecraft stands at the same distance from the Earth, and the gravitational force may be considered constant (i.e., the distance from the Earth, r_{b}, is constant). This observation allows to decouple the orbital mechanics, described by $\boldsymbol{v}_{\mathrm{b_g}}$, from the local dynamics, described by $\delta\boldsymbol{v}_{\mathrm{b}}$. Indeed, since the gravity force is considered constant, the dynamics of $\delta\boldsymbol{v}_{\mathrm{b}}$ is independent since the changes in r_{b} due to this perturbation are not taken into account in $\boldsymbol{f}_{\mathrm{g}}$. The two problems can be decoupled as follows:

$$\begin{aligned} m_{\mathrm{b}}\, \dot{\boldsymbol{v}}_{\mathrm{b_g}} &= \boldsymbol{f}_{\mathrm{g}} \\ m_{\mathrm{b}}\, \delta\dot{\boldsymbol{v}}_{\mathrm{b}} &= \boldsymbol{f}_{\mathrm{dist}} + \boldsymbol{f}_{\mathrm{c}} \end{aligned}$$

This hypothesis holds as long as the translational motion induced by $\delta\boldsymbol{v}_{\mathrm{b}}$ is small compared to the distance from the Earth. Since the thesis scenario is the capture of a tumbling target within reach, the resulting motion of the chaser in translation will be negligible compared to its distance from the ground. No major maneuvers are considered to reach the target's orbit, but rather local translations and rotations around it to compensate for the tumbling motion. It means that the dynamics in weightlessness will be considered for the space robot in the sequel (Dubowsky and Papadopoulos, 1993). The model in (6.47) is thus subject to the control efforts on the spacecraft and at the joints, but not to the gravity. As a result, the velocity profile of the spacecraft is only the perturbation $\delta\boldsymbol{v}_{\mathrm{b}}$ around the orbital speed $\boldsymbol{v}_{\mathrm{b_g}}$. The free-flying dynamics reads similar to (Nenchev, 2013), as follows:

$$\begin{bmatrix} \mathbf{D}_{\mathrm{bb}} & \mathbf{D}_{\mathrm{ba}} \\ \mathbf{D}_{\mathrm{ba}}^{\top} & \mathbf{D}_{\mathrm{aa}} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_{\mathrm{b}} \\ \ddot{\mathbf{q}}_{\mathrm{a}} \end{bmatrix} + \begin{bmatrix} \mathbf{h}_{\mathrm{b}}(\mathbf{q}, \dot{\mathbf{q}}) \\ \mathbf{h}_{\mathrm{a}}(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_{\mathrm{c}} + \boldsymbol{w}_{\mathrm{dist}} \\ \boldsymbol{\tau}_{\mathrm{a}} \end{bmatrix} \tag{6.58}$$

where no external efforts are applied at the end-effectors before the capture, and the gravity gradient is included in the disturbance torques of $\boldsymbol{w}_{\mathrm{dist}}$.

Conclusion

Through this chapter, the modeling of a space robot was derived by extending the previous DeNOC algorithms to a moving base. The on-orbit dynamics was considered from a very general point of view, but will not be included in the next simulations. The resulting efforts

are only considered as external disturbances acting on the spacecraft, and will be taken into account by the control law. Thanks to the common framework used to describe the kinematics and dynamics of rigid and flexible segments, the space robot model was derived in a general form. Indeed, it can account for several chain-like appendages, like manipulators, but also for flexible solar panels or antennas. The only assumption was its “star shape”. The control and the path planning of this complex system will be presented in the following part.

CHAPTER 7 PATH PLANNING FOR DEBRIS CAPTURE

The present chapter derives the path planning method used to perform the on-orbit capture of a tumbling debris by a space robot. The trajectory of the target is first described based on the rigid laws of motion, applied in conditions of weightlessness. Its position lies on a sphere but may reach large variations depending on the tumbling rate of the debris and of its size. In the sequel, the path planner is based on a general optimal criteria to match the position, speed and acceleration of the target point at the instant of capture. This method aims at reducing as much as possible the discontinuous commands sent to the chaser, in order to smooth its capture trajectory and make it more robust to dynamic uncertainties on the debris. The trajectory generation is adapted from (Aghili, 2009a) by adding the acceleration as a new variable state, to ensure its continuity at capture.

7.1 Target Motion

As mentioned in the literature review, the name "debris" denotes the whole spacecraft to be captured, while the "target point" represents the point that the end-effector of the chaser will physically grasp. The debris is considered as a rigid body floating freely in space, in the absence of external efforts and disturbances. This hypothesis does not fully hold in the real case where mainly the atmospheric drag would disturb this dynamics, and where the relative orbital motion w.r.t. the chaser would be considered through the Clohessy-Wiltshire equations (Fehse, 2008). Therefore, the path planning is assumed here as the guidance of the end-effector, leading to a reference trajectory that would be corrected by on-line measurements provided by cameras or lasers.

7.1.1 Debris dynamics and target point kinematics

The debris is supposed to rotate freely in space and to obey the rigid laws of motion introduced in (4.20). As illustrated in Figure 7.1, a frame $\mathcal{R}_\mathrm{d}$ is rigidly fixed to the debris to describe its motion. Using the quaternion representation, the generalized coordinates and velocities of the debris are given by:

$$\mathbf{q}_\mathrm{d} = \begin{bmatrix} \mathbf{r}_\mathrm{d}^\top & \mathcal{Q}_\mathrm{d}^\top \end{bmatrix}^\top \tag{7.1a}$$

$$\dot{\mathbf{q}}_\mathrm{d} = \begin{bmatrix} \boldsymbol{v}_\mathrm{d}^\top & \boldsymbol{\omega}_\mathrm{d}^\top \end{bmatrix}^\top \tag{7.1b}$$

with $(\mathbf{r}_\mathrm{d},\, \mathcal{Q}_\mathrm{d})$ the CoM location and the quaternion of attitude of the debris w.r.t. the inertial frame $\mathcal{R}_\mathrm{I}$, and $(\boldsymbol{v}_\mathrm{d},\, \boldsymbol{\omega}_\mathrm{d})$ its inertial speed and angular rate.

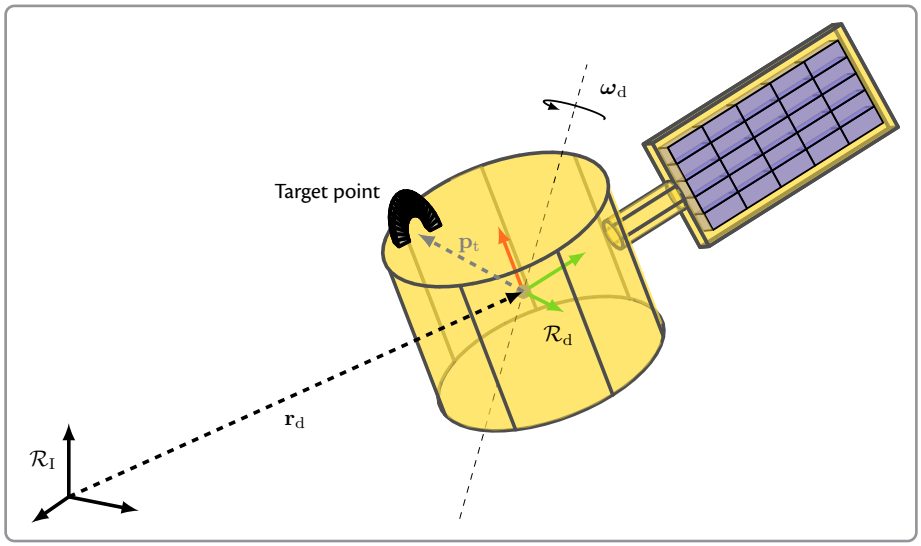


Figure 7.1: Debris kinematics and target point location

From the linear momentum conservation, its expected that the CoM has a constant inertial velocity $\boldsymbol{v}_\mathrm{d} = \boldsymbol{v}_0$. This result derives from the rigid dynamics in translation and rotation, obtained from (4.20) by:

$$m_\mathrm{d}\,\dot{\boldsymbol{v}}_\mathrm{d} = \boldsymbol{f}_\mathrm{d} = \mathbf{0} \tag{7.2a}$$

$$\mathbf{I}_\mathrm{d}\dot{\boldsymbol{\omega}}_\mathrm{d} + \boldsymbol{\omega}_\mathrm{d} \times \mathbf{I}_\mathrm{d}\boldsymbol{\omega}_\mathrm{d} = \boldsymbol{n}_\mathrm{d} = \mathbf{0} \tag{7.2b}$$

where m_d is the mass of the debris, $\mathbf{I}_\mathrm{d}$ its inertia matrix, $(\dot{\boldsymbol{v}}_\mathrm{d}, \dot{\boldsymbol{\omega}}_\mathrm{d})$ the inertial time derivatives of the inertial speed and angular rate, and $(\boldsymbol{f}_\mathrm{d}, \boldsymbol{n}_\mathrm{d})$ the forces and torques applied at the CoM, assumed zero in conditions of weightlessness.

These relations are re-written under the traditional control theory formalism, by isolating the time derivatives of the states in translation and rotation:

$$\dot{\mathbf{r}}_\mathrm{d} = \boldsymbol{v}_\mathrm{d} \tag{7.3a}$$

$$\dot{\boldsymbol{v}}_\mathrm{d} = \mathbf{0} \tag{7.3b}$$

$$\dot{\mathcal{Q}}_\mathrm{d} = \frac{1}{2}\,\mathcal{Q}_\mathrm{d} \otimes \begin{bmatrix} 0 \\ {}^{(\mathrm{d})}\boldsymbol{\omega}_\mathrm{d} \end{bmatrix} \tag{7.3c}$$

$$\dot{\boldsymbol{\omega}}_\mathrm{d} = -\left(\boldsymbol{\omega}_\mathrm{d} \times \mathbf{I}_\mathrm{d}\boldsymbol{\omega}_\mathrm{d}\right) \triangleq \boldsymbol{\phi}(\boldsymbol{\omega}_\mathrm{d}) \tag{7.3d}$$

where $\otimes$ denotes the quaternion product, and the relation between the time derivative of a quaternion and the corresponding angular rate in the body-fixed frame is given in (A.11).

It is worth noticing that the classic "$\dot{\cdot}$" is used for this latter, since there is not a notion of frame of expression.

In the following, the time derivative of the angular rate is denoted by $\boldsymbol{\phi}(\boldsymbol{\omega}_\mathrm{d})$, and holds in both inertial and local frames since the angular rate satisfies: $\dot{\boldsymbol{\omega}} = \mathring{\boldsymbol{\omega}}$. This system of equations in (7.3) can then be summarized by:

$$\dot{\boldsymbol{\chi}}_\mathrm{d} = f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d}) \tag{7.4}$$

with the debris states merging the generalized coordinates and velocities as:

$$\boldsymbol{\chi}_\mathrm{d} = \begin{bmatrix} \mathbf{q}_\mathrm{d}^\top & \dot{\mathbf{q}}_\mathrm{d}^\top \end{bmatrix}^\top \tag{7.5}$$

Using the debris dynamics, the target point trajectory is obtained by deriving its inertial position $\mathbf{r}_\mathrm{t}$, knowing its relative position $\mathbf{p}_\mathrm{t}$ w.r.t. $\mathcal{R}_\mathrm{d}$, as illustrated in Figure 7.1. The computation of the position, speed, acceleration and jerk[1] of the target point are needed to solve the optimal path planning problem afterwards. They read as follows by applying the simple rules of time differentiation in a moving frame:

$$\mathbf{r}_\mathrm{t} = \mathbf{r}_\mathrm{d} + \mathbf{p}_\mathrm{t} \tag{7.6a}$$
$$\dot{\mathbf{r}}_\mathrm{t} = \boldsymbol{v}_\mathrm{d} + \boldsymbol{\omega}_\mathrm{d} \times \mathbf{p}_\mathrm{t} \tag{7.6b}$$
$$\ddot{\mathbf{r}}_\mathrm{t} = \boldsymbol{\phi}(\boldsymbol{\omega}_\mathrm{d}) \times \mathbf{p}_\mathrm{t} + \boldsymbol{\omega}_\mathrm{d} \times (\boldsymbol{\omega}_\mathrm{d} \times \mathbf{p}_\mathrm{t}) \tag{7.6c}$$
$$\dddot{\mathbf{r}}_\mathrm{t} = \dot{\boldsymbol{\phi}}(\boldsymbol{\omega}_\mathrm{d}) \times \mathbf{p}_\mathrm{t} + 2\,\boldsymbol{\phi}(\boldsymbol{\omega}_\mathrm{d}) \times (\boldsymbol{\omega}_\mathrm{d} \times \mathbf{p}_\mathrm{t}) + \boldsymbol{\omega}_\mathrm{d} \times (\boldsymbol{\phi}(\boldsymbol{\omega}_\mathrm{d}) \times \mathbf{p}_\mathrm{t}) - (\boldsymbol{\omega}_\mathrm{d}^\top \boldsymbol{\omega}_\mathrm{d})\, \boldsymbol{\omega}_\mathrm{d} \times \mathbf{p}_\mathrm{t} \tag{7.6d}$$

where the inertial acceleration is zero, $\dot{\boldsymbol{v}}_\mathrm{d} = \mathbf{0}$, and the last term of the RHS in (7.6d) is a triple cross product with $\boldsymbol{\omega}_\mathrm{d}$, that is developed according to the relation:

$$\mathbf{a} \times \mathbf{b} \times \mathbf{c} = (\mathbf{a}^\top \mathbf{c})\, \mathbf{b} - (\mathbf{a}^\top \mathbf{b})\, \mathbf{c} \tag{7.7}$$

In addition, the inertial time derivative of the function $\boldsymbol{\phi}$ is obtained as:

$$\begin{aligned} \dot{\boldsymbol{\phi}}(\boldsymbol{\omega}_\mathrm{d}) &= \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{\mathcal{R}_\mathrm{d}} (\boldsymbol{\phi}(\boldsymbol{\omega}_\mathrm{d})) + \boldsymbol{\omega}_\mathrm{d} \times \boldsymbol{\phi}(\boldsymbol{\omega}_\mathrm{d}) \\ &= -\mathbf{I}_\mathrm{d}^{-1} \left(\mathring{\boldsymbol{\omega}}_\mathrm{d} \times \mathbf{I}_\mathrm{d} \boldsymbol{\omega}_\mathrm{d} + \boldsymbol{\omega}_\mathrm{d} \times \mathbf{I}_\mathrm{d} \mathring{\boldsymbol{\omega}}_\mathrm{d}\right) + \boldsymbol{\omega}_\mathrm{d} \times \boldsymbol{\phi}(\boldsymbol{\omega}_\mathrm{d}) \\ &= -\mathbf{I}_\mathrm{d}^{-1} \left(\boldsymbol{\phi}(\boldsymbol{\omega}_\mathrm{d}) \times \mathbf{I}_\mathrm{d} \boldsymbol{\omega}_\mathrm{d} + \boldsymbol{\omega}_\mathrm{d} \times \mathbf{I}_\mathrm{d} \boldsymbol{\phi}(\boldsymbol{\omega}_\mathrm{d})\right) + \boldsymbol{\omega}_\mathrm{d} \times \boldsymbol{\phi}(\boldsymbol{\omega}_\mathrm{d}) \end{aligned} \tag{7.8}$$

[1]This term denotes the third time derivative of a position and is often used in robotics to generate smooth motion.

The inertial position, speed and acceleration of the target point are known and will be used in the optimal problem as final conditions to ensure that the end-effector can synchronize with it in either position, speed, acceleration or jerk.

7.2 Optimal Robot Guidance

The computation of the optimal capture trajectory may be led by many different means, from the analytic solutions on reduced problems to the sophisticated numerical methods to account for a great number of constraints. In the sequel, the path planning of the end-effector is performed based on the analytic solution of a reduced optimization problem derived from the application of the Pontryagin's minimum principle (Athans and Falb, 1966; Bryson and Ho, 1975). The optimization criteria is taken from the work of F. Aghili in (Aghili, 2009a), where the optimal capture trajectory guarantees the continuity of the end-effector position and speed with the target point ones. This problem is augmented here to extend this continuity to the acceleration, in order **to allow the end-effector to smoothly switch from the capture trajectory to the natural trajectory of the target point**.The grapple fixture can then be tracked for a short time period, before grasping it in a safer way.

7.2.1 Optimal problem statement

The optimization problem is first put in a general form, before refining it for each set of matching conditions at capture. Two main constraints are considered: the system dynamics and these final conditions. The first one is given by merging the dynamics of the debris and the ones of the path planner. The second one is only applied at the final time t_f, and corresponds to the capture conditions: either with the continuity in position, speed or acceleration.

As a general notation, the state variables and inputs required by the path planner are denoted, respectively, by $\boldsymbol{\chi}_{\mathrm{h}}$ and $\mathbf{u}_{\mathrm{h}}$, where the subscript "h" stands for the trajectory of the chaser's end-effector, also called "hand" in a more casual way. These quantities are merged with the debris variables to define the global system states and inputs by $\boldsymbol{\chi} = (\boldsymbol{\chi}_{\mathrm{d}}, \boldsymbol{\chi}_{\mathrm{h}})$ and $\mathbf{u} = \mathbf{u}_{\mathrm{h}}$, since the debris is in weightlessness and does not have any input signal. The corresponding final conditions are denoted by $\boldsymbol{\psi}(\boldsymbol{\chi}(t_f)) = 0$, and are assumed not to have a direct dependency upon time.

With these notations, the path planner lies on the following set of dynamic equations:

$$\dot{\boldsymbol{\chi}}_{\mathrm{h}} = f_{\mathrm{h}}(\boldsymbol{\chi}_{\mathrm{h}}, \mathbf{u}) \tag{7.9}$$

that are merged with the debris dynamics in (7.4) to describe the dynamic constraint by:

$$\dot{\boldsymbol{\chi}} = \begin{bmatrix} f_{\mathrm{d}}(\boldsymbol{\chi}_{\mathrm{d}}) \\ f_{\mathrm{h}}(\boldsymbol{\chi}_{\mathrm{h}}, \mathbf{u}_{\mathrm{h}}) \end{bmatrix} \triangleq f(\boldsymbol{\chi}, \mathbf{u}) \tag{7.10}$$

The general form of the optimization problem can now be written as:

$$\min_{\mathbf{r}_{\mathrm{h}}(t)} \; J = \int_0^{t_f} L(\boldsymbol{\chi}, \mathbf{u}) \mathrm{d}t \tag{7.11a}$$

$$\text{subject to} \quad \begin{cases} \dot{\boldsymbol{\chi}} = f(\boldsymbol{\chi}, \mathbf{u}) \\ \boldsymbol{\psi}(\boldsymbol{\chi}(t_f)) = \mathbf{0} \end{cases} \tag{7.11b}$$

where the Lagrangian notation is used to denote the cost function integrated in the criterion by $L(\boldsymbol{\chi}, \mathbf{u})$. The corresponding Hamiltonian is defined as follows, where $\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_{\mathrm{d}}^\top & \boldsymbol{\lambda}_{\mathrm{h}}^\top \end{bmatrix}^\top$ are the Lagrange multipliers related to the constraints:

$$H(\boldsymbol{\chi}, \mathbf{u}, \boldsymbol{\lambda}) = L(\boldsymbol{\chi}, \mathbf{u}) + \boldsymbol{\lambda}_{\mathrm{d}}^\top f_{\mathrm{d}}(\boldsymbol{\chi}_{\mathrm{d}}) + \boldsymbol{\lambda}_{\mathrm{h}}^\top f_{\mathrm{h}}(\boldsymbol{\chi}_{\mathrm{h}}, \mathbf{u}_{\mathrm{h}}) \tag{7.12}$$

Applying the Pontryagin's minimum principle on the optimization problem in (7.11a), the following set of equations is used to describe the optimal solution for a free final time (Athans and Falb, 1966; Bryson and Ho, 1975). The capture trajectory will be defined by the optimal values of the input $\mathbf{u}^*(t)$, the states $\boldsymbol{\chi}^*(t)$, the co-states $\boldsymbol{\lambda}^*(t)$ and $\boldsymbol{\nu}^*$, and the Hamiltonian

$H^*(t)$, which are solution of:

$$\dot{\boldsymbol{\chi}} = f(\boldsymbol{\chi}, \mathbf{u}) \tag{7.13a}$$

$$\dot{\boldsymbol{\lambda}} = -\left(\frac{\partial f}{\partial \boldsymbol{\chi}}\right)^\top \boldsymbol{\lambda} - \left(\frac{\partial L}{\partial \boldsymbol{\chi}}\right)^\top \tag{7.13b}$$

$$\left(\frac{\partial H}{\partial \mathbf{u}}\right)^\top = \left(\frac{\partial f}{\partial \mathbf{u}}\right)^\top \boldsymbol{\lambda} + \left(\frac{\partial L}{\partial \mathbf{u}}\right)^\top = \mathbf{0} \tag{7.13c}$$

$$\boldsymbol{\chi}(0) = \boldsymbol{\chi}_0 \tag{7.13d}$$

$$\boldsymbol{\lambda}(t_f) = \left(\boldsymbol{\nu}^\top \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{\chi}}\right)^\top_{t=t_f} \tag{7.13e}$$

$$H(t_f) = 0 \tag{7.13f}$$

$$\boldsymbol{\psi}\left(\boldsymbol{\chi}(t_f)\right) = 0 \tag{7.13g}$$

where the equations represent, in order of appearance, the states dynamics in (7.13a), the co-states dynamics in (7.13b), the implicit definition of the optimal input in (7.13c), the initial conditions on the states in (7.13d), the *transversality condition* with the final condition on the co-states in (7.13e), the optimal Hamiltonian at t_f in (7.13f), and the final conditions on the states in (7.13g).

Thanks to the decoupled nature of the system dynamics in (7.10), the first three equations can be further developed, as follows:

$$\begin{bmatrix} \dot{\boldsymbol{\chi}}_\mathrm{d} \\ \dot{\boldsymbol{\chi}}_\mathrm{h} \end{bmatrix} = \begin{bmatrix} f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d}) \\ f_\mathrm{h}(\boldsymbol{\chi}_\mathrm{h}, \mathbf{u}_\mathrm{h}) \end{bmatrix} \tag{7.14a}$$

$$\begin{bmatrix} \dot{\boldsymbol{\lambda}}_\mathrm{d} \\ \dot{\boldsymbol{\lambda}}_\mathrm{h} \end{bmatrix} = -\begin{bmatrix} \left(\frac{\partial f_\mathrm{d}}{\partial \boldsymbol{\chi}_\mathrm{d}}\right)^\top & \mathbf{0} \\ \mathbf{0} & \left(\frac{\partial f_\mathrm{h}}{\partial \boldsymbol{\chi}_\mathrm{h}}\right)^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_\mathrm{d} \\ \boldsymbol{\lambda}_\mathrm{h} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \left(\frac{\partial L}{\partial \boldsymbol{\chi}_\mathrm{h}}\right)^\top \end{bmatrix} \tag{7.14b}$$

$$\left(\frac{\partial H}{\partial \mathbf{u}}\right)^\top = \left(\frac{\partial f_\mathrm{h}}{\partial \mathbf{u}_\mathrm{h}}\right)^\top \boldsymbol{\lambda}_\mathrm{h} + \left(\frac{\partial L}{\partial \mathbf{u}_\mathrm{h}}\right)^\top = \mathbf{0} \tag{7.14c}$$

where an additional hypothesis is made by assuming that L only depends on the path planner states in $\boldsymbol{\chi}_\mathrm{h}$, and not on the debris ones in $\boldsymbol{\chi}_\mathrm{d}$.

In the sequel, this optimal problem is solved analytically for three different set of final conditions in position, speed and acceleration. Due to these latter, the criterion and the Hamiltonian definitions change, and the whole set of equations is updated accordingly.

7.2.2 Optimal trajectories

To compare the trajectories resulting from the acceleration continuity with the position and speed continuities, three different sets of final conditions are considered thereafter. For each of them, the trajectory is assumed to be generated by an input corresponding to the time derivative of the most differentiated signal in the final conditions. It means that the path planner input is either the speed, the acceleration or the jerk of the inertial position of the trajectory.

The set of equations defining the optimal quantities is always solved in the same fashion because similar cost functions and final conditions are defined. It reads as follows:

Step 1 Using only the state dynamics of the path planner in (7.14a), the optimal input $\mathbf{u}_\mathrm{h}$ is expressed as a time derivative of the position $\mathbf{r}_\mathrm{h}$. Similarly, the co-states of the path planner $\boldsymbol{\lambda}_\mathrm{h}$ are expressed as functions of $\mathbf{u}_\mathrm{h}$ by (7.14c);

Step 2 Replacing these co-states $\boldsymbol{\lambda}_\mathrm{h}$ by $\mathbf{u}_\mathrm{h}$ in their dynamics in (7.14b), and then replacing $\mathbf{u}_\mathrm{h}$ by $\mathbf{r}_\mathrm{h}$, an *Ordinary Differential Equation (ODE)* appears to define the optimal trajectory shape;

Step 3 The resulting ODE is solved analytically and is parameterized by a set of variables $\{\mathbf{k}_i\}$;

Step 4 The initial and final conditions in (7.13d) and (7.13g) are used to define symbolically the parameters $\{\mathbf{k}_i\}$ as functions of the final time t_f;

Step 5 The optimal Hamiltonian in (7.13f) provides an implicit equation to define the final time. This relation is derived by replacing $\mathbf{u}_\mathrm{h}$ with $\mathbf{r}_\mathrm{h}$ using **step 1**, by simplifying various terms with the final conditions in (7.13g), and eventually by using the following trick from (Aghili, 2009a) to express the state derivative of $\mathbf{r}_\mathrm{t}$ w.r.t. $\boldsymbol{\chi}_\mathrm{d}$ as a function of its speed:

$$\dot{\mathbf{r}}_\mathrm{t} = \left(\frac{\partial \mathbf{r}_\mathrm{t}}{\partial \boldsymbol{\chi}_\mathrm{d}}\right)\left(\frac{\partial \boldsymbol{\chi}_\mathrm{d}}{\partial t}\right) = \left(\frac{\partial \mathbf{r}_\mathrm{t}}{\partial \boldsymbol{\chi}_\mathrm{d}}\right) f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d})$$

Position Matching When the position is the only final condition at the instant of capture, the trajectory is generated by a low order derivative. It is sufficient to define the path planner input as the speed profile to fulfill only one final condition at the capture. The path planner input and the corresponding vector $\boldsymbol{\psi}$ are defined as follows:

$$\dot{\mathbf{r}}_\mathrm{h} = \mathbf{u}_\mathrm{h} \qquad \text{and} \qquad \underbrace{\left[\ \mathbf{r}_\mathrm{h}(t_f) - \mathbf{r}_\mathrm{t}(t_f)\ \right]}_{\boldsymbol{\psi}(\boldsymbol{\chi}_\mathrm{h}(t_f))} = \mathbf{0}_{3\times 1} \tag{7.15}$$

Therefore, the path planner states, the Lagrangian, and the Hamiltonian are defined by:

$$\boldsymbol{\chi}_{\mathrm{h}} = \left[\ \mathbf{r}_{\mathrm{h}}\ \right] \tag{7.16a}$$

$$\dot{\boldsymbol{\chi}}_{\mathrm{h}} = \left[\ \mathbf{u}_{\mathrm{h}}\ \right] (= f_{\mathrm{h}}(\boldsymbol{\chi}_{\mathrm{h}}, \mathbf{u}_{\mathrm{h}})) \tag{7.16b}$$

$$L(\boldsymbol{\chi}_{\mathrm{h}}, \mathbf{u}_{\mathrm{h}}) = 1 + w_u \left\|\mathbf{u}_{\mathrm{h}}\right\|^2 \tag{7.16c}$$

$$H(\boldsymbol{\chi}, \mathbf{u}_{\mathrm{h}}) = 1 + w_u \left\|\mathbf{u}_{\mathrm{h}}\right\|^2 + \boldsymbol{\lambda}_{\mathrm{d}}^{\top} f_{\mathrm{d}}(\boldsymbol{\chi}_{\mathrm{d}}) + \boldsymbol{\lambda}_{\mathrm{h}}^{\top} \mathbf{u}_{\mathrm{h}} \tag{7.16d}$$

where the weight w_u is introduced to manage the trade-off between minimal time and consumption objectives.

First of all, **step 1** is fulfilled by the previous definition of $\mathbf{u}_{\mathrm{h}}$ in (7.16b), and by the development of the equation (7.14c) with the Hamiltonian in (7.16d), as follows:

$$\left(\frac{\partial H}{\partial \mathbf{u}}\right)^{\top} = \underbrace{\left(\frac{\partial f_{\mathrm{h}}}{\partial \mathbf{u}_{\mathrm{h}}}\right)^{\top}}_{=\mathbf{I}_3} \boldsymbol{\lambda}_{\mathrm{h}} + \underbrace{\left(\frac{\partial L}{\partial \mathbf{u}_{\mathrm{h}}}\right)^{\top}}_{=2\,w_u\,\mathbf{u}_{\mathrm{h}}} = \mathbf{0}_{3\times 1}$$

leading to:

$$\boldsymbol{\lambda}_{\mathrm{h}} = -2\, w_u\, \mathbf{u}_{\mathrm{h}} \tag{7.17}$$

Turning then to the co-states dynamics of the path planner in (7.14b), one obtains that:

$$\dot{\boldsymbol{\lambda}}_{\mathrm{h}} = -\left(\frac{\partial f_{\mathrm{h}}}{\partial \boldsymbol{\chi}_{\mathrm{h}}}\right)^{\top} \boldsymbol{\lambda}_{\mathrm{h}} - \left(\frac{\partial L}{\partial \boldsymbol{\chi}_{\mathrm{h}}}\right)^{\top} = \mathbf{0}_{3\times 1}$$

Implying that $\boldsymbol{\lambda}_{\mathrm{h}} = \boldsymbol{\lambda}_{\mathrm{h}}(0)$ is a constant. Replacing then $\mathbf{u}_{\mathrm{h}}$ by (7.16b) into (7.17), the differential equation of **steps 2-3** is given and solved by:

$$\dot{\mathbf{r}}_{\mathrm{h}} = -\frac{\boldsymbol{\lambda}_{\mathrm{h}}(0)}{2\, w_u} \qquad \Longrightarrow \qquad \mathbf{r}_{\mathrm{h}}(t) = \mathbf{k}_0 + \mathbf{k}_1\, t \tag{7.18}$$

Denoting the initial condition by $\mathbf{r}_{\mathrm{h}}(0) = \mathbf{r}_{\mathrm{h}_0}$ and using the final condition in (7.15), an algebraic manipulation allows to performs the **step 4** as:

$$\mathbf{k}_0 = \mathbf{r}_{\mathrm{h}_0} \qquad \text{and} \qquad \mathbf{k}_1 = \frac{\mathbf{r}_{\mathrm{t}}(t_f) - \mathbf{r}_{\mathrm{h}_0}}{t_f} \tag{7.19}$$

The determination of the optimal time of capture in **step 5** is now derived under an implicit

equation. Coming back to (7.13f), it yields:

$$H(t_f) = 0 = \left(1 + w_u\, \mathbf{u}_\mathrm{h}^\top \mathbf{u}_\mathrm{h} + \boldsymbol{\lambda}_\mathrm{d}^\top\, f_\mathrm{d} + \boldsymbol{\lambda}_\mathrm{h}^\top\, \mathbf{u}_\mathrm{h}\right)_{t_f} \tag{7.20}$$

$$= 1 + w_u\, \dot{\mathbf{r}}_\mathrm{h}(t_f)^\top \dot{\mathbf{r}}_\mathrm{h}(t_f) + \left(\boldsymbol{\lambda}_\mathrm{d}^\top\, f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d})\right)_{t_f} - 2w_u\, \dot{\mathbf{r}}_\mathrm{h}(t_f)^\top\, \dot{\mathbf{r}}_\mathrm{h}(t_f) \tag{7.21}$$

where the remaining unknown $\boldsymbol{\lambda}_\mathrm{d}(t_f)$ is obtained by the *transversality condition* in (7.13e):

$$\begin{bmatrix} \boldsymbol{\lambda}_\mathrm{d} \\ \boldsymbol{\lambda}_\mathrm{h} \end{bmatrix}_{t_f} = \left(\nu_\psi^\top \begin{bmatrix} \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{\chi}_\mathrm{d}} & \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{\chi}_\mathrm{h}} \end{bmatrix}\right)^\top_{t_f}$$

$$= \begin{bmatrix} -\left(\frac{\partial \mathbf{r}_\mathrm{t}}{\partial \boldsymbol{\chi}_\mathrm{d}}\right)^\top \nu_\psi \\ \nu_\psi \end{bmatrix}_{t_f}$$

which leads to $\nu_\psi = \boldsymbol{\lambda}_\mathrm{h}(t_f)$. The following relation is then obtained, using the trick mentioned above, replacing $\boldsymbol{\lambda}_\mathrm{h}$ by $\mathbf{u}_\mathrm{h}$ with (7.17), and finally by replacing $\mathbf{u}_\mathrm{h}$ by its definition with $\dot{\mathbf{r}}_\mathrm{h}$:

$$\left(\boldsymbol{\lambda}_\mathrm{d}^\top\, f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d})\right)_{t_f} = \boldsymbol{\lambda}_\mathrm{h}(t_f)^\top \left(-\frac{\partial \mathbf{r}_\mathrm{t}}{\partial \boldsymbol{\chi}_\mathrm{d}}\, f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d})\right) = 2w_u\, \dot{\mathbf{r}}_\mathrm{h}(t_f)^\top\, \dot{\mathbf{r}}_\mathrm{t}(t_f)$$

Eventually, the implicit equation defining the optimal time is given by:

$$\boxed{1 - w_u\, \dot{\mathbf{r}}_\mathrm{h}(t_f)^\top \dot{\mathbf{r}}_\mathrm{h}(t_f) + 2w_u\, \dot{\mathbf{r}}_\mathrm{h}(t_f)^\top\, \dot{\mathbf{r}}_\mathrm{t}(t_f) = 0} \tag{7.22}$$

This equation is solved numerically by replacing the analytic expression of the speed $\dot{\mathbf{r}}_\mathrm{h}$ with the time derivative of (7.18) evaluated at t_f, and propagating the debris dynamics on a large time horizon in order to evaluate the value of $\dot{\mathbf{r}}_\mathrm{t}$ at any time t_f.

An example of the capture trajectory is presented in Figure 7.2. Using this guidance scheme, the debris would be captured very roughly since a straight line is followed toward the target point. Hence, two jumps appear on the speed and on the acceleration profile, and are necessary to keep following the natural target trajectory. This optimal result is derived for a debris whose kinematic and dynamic parameters ($\mathbf{I}_\mathrm{d}$, $\mathbf{p}_\mathrm{t}$) are given in Appendix D.2. The

following sets of initial conditions were used on the target and on the capture trajectory:

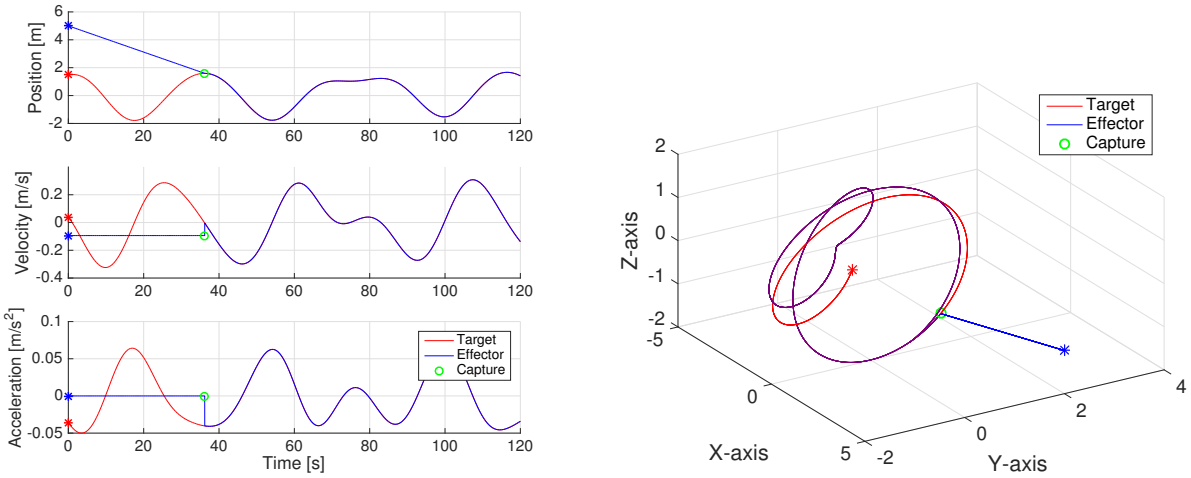
$$\mathbf{r}_{\mathrm{d}_0} = \mathbf{0}_{3\times1} \tag{7.23}$$

$$\boldsymbol{v}_{\mathrm{d}_0} = \mathbf{0}_{3\times1} \tag{7.24}$$

$$\mathcal{Q}_{\mathrm{d}_0} = \left[\begin{array}{cc} \cos(0.5) & -\sin(0.5)\left[\begin{array}{ccc} 0 & 0 & 1 \end{array}\right] \end{array}\right]^{\top} \tag{7.25}$$

$$\boldsymbol{\omega}_{\mathrm{d}_0} = \left[\begin{array}{ccc} -0.15 & 0.05 & -0.08 \end{array}\right] \tag{7.26}$$

$$\mathbf{r}_{\mathrm{h}_0} = \left[\begin{array}{ccc} 5 & 2 & -1 \end{array}\right] \tag{7.27}$$



(a) (b)

Figure 7.2: Example of an optimal capture trajectory by matching the position;
(a) X-axis components of the trajectory; (b) 3D capture trajectory

Position/Speed Matching In order to avoid any impact at the capture, the end-effector speed must match the target one. It results in an additional constraint that forces to generate the trajectory by its acceleration profile. Indeed, more DoF must be added to the optimal solution in order to fulfill these extra boundary conditions. This case is mainly reproduced from (Aghili, 2009a), and will be extended in the following to account for the acceleration continuity.

To ensure the speed continuity at capture, the input becomes the acceleration of the trajectory:

$$\ddot{\mathbf{r}}_{\mathrm{h}} = \mathbf{u}_{\mathrm{h}} \qquad \text{and} \qquad \underbrace{\left[\begin{array}{c} \mathbf{r}_{\mathrm{h}}(t_f) - \mathbf{r}_{\mathrm{t}}(t_f) \\ \dot{\mathbf{r}}_{\mathrm{h}}(t_f) - \dot{\mathbf{r}}_{\mathrm{t}}(t_f) \end{array}\right]}_{\psi(\boldsymbol{\chi}_{\mathrm{h}}(t_f))} = \mathbf{0}_{6\times1} \tag{7.28}$$

The problem variables are re-defined as follows:

$$\boldsymbol{\chi}_{\mathrm{h}} = \begin{bmatrix} \mathbf{r}_{\mathrm{h}} \\ \boldsymbol{v}_{\mathrm{h}} \end{bmatrix} \tag{7.29a}$$

$$\dot{\boldsymbol{\chi}}_{\mathrm{h}} = \begin{bmatrix} \boldsymbol{v}_{\mathrm{h}} \\ \mathbf{u}_{\mathrm{h}} \end{bmatrix} (= f_{\mathrm{h}}(\boldsymbol{\chi}_{\mathrm{h}}, \mathbf{u}_{\mathrm{h}})) \tag{7.29b}$$

$$L(\boldsymbol{\chi}_{\mathrm{h}}, \mathbf{u}_{\mathrm{h}}) = 1 + w_1 \|\boldsymbol{v}_{\mathrm{h}}\|^2 + w_u \|\mathbf{u}_{\mathrm{h}}\|^2 \tag{7.29c}$$

$$H(\boldsymbol{\chi}, \mathbf{u}_{\mathrm{h}}) = 1 + w_1 \|\boldsymbol{v}_{\mathrm{h}}\|^2 + w_u \|\mathbf{u}_{\mathrm{h}}\|^2 + \boldsymbol{\lambda}_{\mathrm{d}}^{\top} f_{\mathrm{d}}(\boldsymbol{\chi}_{\mathrm{d}}) + \boldsymbol{\lambda}_{\mathrm{h}_1}^{\top} \boldsymbol{v}_{\mathrm{h}} + \boldsymbol{\lambda}_{\mathrm{h}_2}^{\top} \mathbf{u}_{\mathrm{h}} \tag{7.29d}$$

The problem is solved exactly in the same fashion, following the steps introduced above. Again, **step 1** lies on the definition of $\mathbf{u}_{\mathrm{h}}$ in (7.29b), and on the development of (7.14c) to obtain:

$$\begin{aligned} \left(\frac{\partial H}{\partial \mathbf{u}}\right)^{\top} = \mathbf{0}_{3\times 1} &= \left(\frac{\partial f_{\mathrm{h}}}{\partial \mathbf{u}_{\mathrm{h}}}\right)^{\top} \boldsymbol{\lambda}_{\mathrm{h}} + \left(\frac{\partial L}{\partial \mathbf{u}_{\mathrm{h}}}\right)^{\top} \\ &= \left(\begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{I}_3 \end{bmatrix}\right) \boldsymbol{\lambda}_{\mathrm{h}} + 2\, w_u\, \mathbf{u}_{\mathrm{h}} \\ &= \boldsymbol{\lambda}_{\mathrm{h}_2} + 2\, w_u\, \mathbf{u}_{\mathrm{h}} \end{aligned}$$

leading to the similar result:

$$\boldsymbol{\lambda}_{\mathrm{h}_2} = -2\, w_u\, \mathbf{u}_{\mathrm{h}} \tag{7.30}$$

The co-states dynamics are developed as:

$$\begin{aligned} \dot{\boldsymbol{\lambda}}_{\mathrm{h}} &= -\left(\frac{\partial f_{\mathrm{h}}}{\partial \boldsymbol{\chi}_{\mathrm{h}}}\right)^{\top} \boldsymbol{\lambda}_{\mathrm{h}} - \left(\frac{\partial L}{\partial \boldsymbol{\chi}_{\mathrm{h}}}\right)^{\top} \\ &= -\begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{I}_3 \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix}^{\top} \boldsymbol{\lambda}_{\mathrm{h}} - \begin{bmatrix} \mathbf{0}_{1\times 3} & 2w_1\, \boldsymbol{v}_{\mathrm{h}}^{\top} \end{bmatrix}^{\top} \\ &= \begin{bmatrix} \mathbf{0}_{3\times 1} \\ -\boldsymbol{\lambda}_{\mathrm{h}_1} - 2w_1\, \boldsymbol{v}_{\mathrm{h}} \end{bmatrix} \end{aligned}$$

implying that $\boldsymbol{\lambda}_{\mathrm{h}_1} = \boldsymbol{\lambda}_{\mathrm{h}_1}(0)$ is still a constant. Replacing $\mathbf{u}_{\mathrm{h}}$ by (7.28) in (7.30), and differentiating it with respect to time, the differential equation for $\mathbf{r}_{\mathrm{h}}$ is derived as follows:

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\left(\ddot{\mathbf{r}}_{\mathrm{h}} - \frac{w_1}{w_u}\mathbf{r}_{\mathrm{h}}\right) = \mathbf{0}_{3\times 1} \tag{7.31}$$

Defining $\sigma = w_1/w_u > 0$, the previous ODE is first integrated twice to obtain the RHS under a polynomial form: $\alpha_0 + \alpha_1\, t$. The homogeneous and particular solutions of this problem are

then obtained by:

$$\mathbf{r}_{\mathrm{h}}^{H} = \mathbf{k}_2\, e^{\sigma\, t} + \mathbf{k}_3\, e^{-\sigma\, t}$$
$$\mathbf{r}_{\mathrm{h}}^{P} = \mathbf{k}_0 + \mathbf{k}_1\, t$$

leading to the following general solution:

$$\boxed{\mathbf{r}_{\mathrm{h}}(t) = \mathbf{k}_0 + \mathbf{k}_1\, t + \mathbf{k}_2\, e^{\sigma\, t} + \mathbf{k}_3\, e^{-\sigma\, t}} \tag{7.32}$$

Using the final condition in (7.28) to obtain the parameters $\{\mathbf{k}_i\}$, **step 4** is solved by the following linear system:

$$\left[\begin{array}{cc|cc} \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \sigma\,\mathbf{I}_3 & -\sigma\,\mathbf{I}_3 \\ \hline \mathbf{I}_3 & t_f\,\mathbf{I}_3 & e^{\sigma\, t_f}\,\mathbf{I}_3 & e^{-\sigma\, t_f}\,\mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \sigma\, e^{\sigma\, t_f}\,\mathbf{I}_3 & -\sigma\, e^{-\sigma\, t_f}\,\mathbf{I}_3 \end{array}\right] \left[\begin{array}{c} \mathbf{k}_0 \\ \mathbf{k}_1 \\ \mathbf{k}_2 \\ \mathbf{k}_3 \end{array}\right] = \left[\begin{array}{c} \mathbf{r}_{\mathrm{h}}(0) \\ \boldsymbol{v}_{\mathrm{h}}(0) \\ \hline \mathbf{r}_{\mathrm{t}}(t_f) \\ \dot{\mathbf{r}}_{\mathrm{t}}(t_f) \end{array}\right] \tag{7.33}$$

As an alternative, if no weight were given to the speed in the cost function L, i.e., if $w_1 = 0$, the ODE would reduce to:

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\,(\ddot{\mathbf{r}}_{\mathrm{h}}) = \mathbf{0}_{3\times1} \tag{7.34}$$

and lead to the following polynomial solution:

$$\mathbf{r}_{\mathrm{h}}(t) = \mathbf{k}_0 + \mathbf{k}_1\, t + \mathbf{k}_2\, t^2 + \mathbf{k}_3\, t^3 \tag{7.35}$$

The corresponding linear system with the initial and final conditions is given by:

$$\left[\begin{array}{cc|cc} \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \hline \mathbf{I}_3 & t_f\,\mathbf{I}_3 & t_f^2\,\mathbf{I}_3 & t_f^3\,\mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & 2t_f\,\mathbf{I}_3 & 3t_f^2\,\mathbf{I}_3 \end{array}\right] \left[\begin{array}{c} \mathbf{k}_0 \\ \mathbf{k}_1 \\ \mathbf{k}_2 \\ \mathbf{k}_3 \end{array}\right] = \left[\begin{array}{c} \mathbf{r}_{\mathrm{h}}(0) \\ \boldsymbol{v}_{\mathrm{h}}(0) \\ \hline \mathbf{r}_{\mathrm{t}}(t_f) \\ \dot{\mathbf{r}}_{\mathrm{t}}(t_f) \end{array}\right] \tag{7.36}$$

For both hypothesis, the previous step can be done analytically or by symbolic computation. The idea is to express the parameters as explicit functions of t_f. The trajectory $\mathbf{r}_{\mathrm{h}}$ can then be expressed as an explicit function of time, in order to derive the implicit equation defining the optimal final time.

Coming back to (7.13f), **step 5** is given by:

$$\begin{aligned} H(t_f) = 0 &= \left(1 + w_1 \left\|\boldsymbol{v}_\mathrm{h}\right\|^2 + w_u \, \mathbf{u}_\mathrm{h}(t_f)^\top \mathbf{u}_\mathrm{h}(t_f) + \boldsymbol{\lambda}_\mathrm{d}(t_f)^\top f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d}) + \boldsymbol{\lambda}_{\mathrm{h}_1}^\top \boldsymbol{v}_\mathrm{h} + \boldsymbol{\lambda}_{\mathrm{h}_2}^\top \mathbf{u}_\mathrm{h}\right)_{t_f} \\ &= \left(1 + w_1 \, \dot{\mathbf{r}}_\mathrm{h}^\top \dot{\mathbf{r}}_\mathrm{h} + w_u \, \ddot{\mathbf{r}}_\mathrm{h} \top \ddot{\mathbf{r}}_\mathrm{h} + \boldsymbol{\lambda}_\mathrm{d}^\top f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d}) + \boldsymbol{\lambda}_{\mathrm{h}_1}^\top \dot{\mathbf{r}}_\mathrm{h} - 2w_u \, \ddot{\mathbf{r}}_\mathrm{h}^\top \ddot{\mathbf{r}}_\mathrm{h}\right)_{t_f} \\ &= 1 + w_1 \, \dot{\mathbf{r}}_\mathrm{h}(t_f)^\top \dot{\mathbf{r}}_\mathrm{h}(t_f) - w_u \, \ddot{\mathbf{r}}_\mathrm{h}(t_f)^\top \ddot{\mathbf{r}}_\mathrm{h}(t_f) + \left(\boldsymbol{\lambda}_\mathrm{d}^\top f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d})\right)_{t_f} + \boldsymbol{\lambda}_{\mathrm{h}_1}^\top \dot{\mathbf{r}}_\mathrm{h}(t_f) \end{aligned}$$

where the remaining unknown $\boldsymbol{\lambda}_\mathrm{d}(t_f)$ is obtained by (7.13e):

$$\begin{bmatrix} \boldsymbol{\lambda}_\mathrm{d} \\ \boldsymbol{\lambda}_\mathrm{h} \end{bmatrix}_{t_f} = \left(\nu_\psi^\top \begin{bmatrix} \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{\chi}_\mathrm{d}} & \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{\chi}_\mathrm{h}} \end{bmatrix}\right)_{t_f}^\top = \begin{bmatrix} -\left(\frac{\partial \mathbf{r}_\mathrm{t}}{\partial \boldsymbol{\chi}_\mathrm{d}}\right)^\top \nu_{\psi_1} - \left(\frac{\partial \dot{\mathbf{r}}_\mathrm{t}}{\partial \boldsymbol{\chi}_\mathrm{d}}\right)^\top \nu_{\psi_2} \\ \nu_{\psi_1} \\ \nu_{\psi_2} \end{bmatrix}_{t_f}$$

which leads to $\nu_{\psi_1} = \boldsymbol{\lambda}_{\mathrm{h}_1}(t_f)$ and $\nu_{\psi_2} = \boldsymbol{\lambda}_{\mathrm{h}_2}(t_f)$. Carrying out the same computations as for the position matching only, and using (7.30), one obtains that:

$$\begin{aligned} \left(\boldsymbol{\lambda}_\mathrm{d}^\top f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d})\right)_{t_f} &= \boldsymbol{\lambda}_{\mathrm{h}_1}(t_f)^\top \left(-\frac{\partial \mathbf{r}_\mathrm{t}}{\partial \boldsymbol{\chi}_\mathrm{d}} f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d})\right) + \boldsymbol{\lambda}_{\mathrm{h}_2}(t_f)^\top \left(-\frac{\partial \dot{\mathbf{r}}_\mathrm{t}}{\partial \boldsymbol{\chi}_\mathrm{d}} f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d})\right) \\ &= -\boldsymbol{\lambda}_{\mathrm{h}_1}(t_f)^\top \dot{\mathbf{r}}_\mathrm{t}(t_f) + 2w_u \, \ddot{\mathbf{r}}_\mathrm{h}(t_f)^\top \ddot{\mathbf{r}}_\mathrm{t}(t_f) \end{aligned}$$

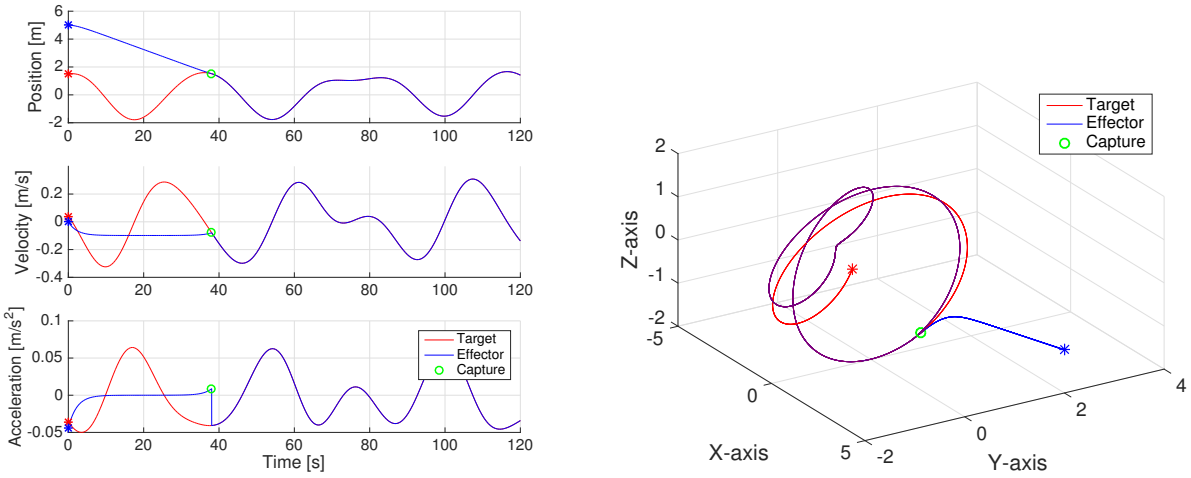
Eventually, simplifying the implicit equation with the final condition on speed in (7.28), the optimal time satisfies:

$$\boxed{1 + w_1 \, \dot{\mathbf{r}}_\mathrm{h}(t_f)^\top \dot{\mathbf{r}}_\mathrm{h}(t_f) - w_u \, \ddot{\mathbf{r}}_\mathrm{h}(t_f)^\top \ddot{\mathbf{r}}_\mathrm{h}(t_f) + 2w_u \, \ddot{\mathbf{r}}_\mathrm{h}(t_f)^\top \ddot{\mathbf{r}}_\mathrm{t}(t_f) = 0} \tag{7.37}$$

This equation is also solved numerically by replacing the speed and acceleration by their analytic expression evaluated at t_f, based on (7.31). The results are illustrated in Figure 7.3 for the same initial condition in (7.23).

It is clearly seen how the trajectory becomes smoother, but at the expense of an acceleration jump even greater at the instant of capture. The advantage of this guidance scheme is to allow a capture without impact, in theory, but its main drawback is to require an instantaneous update of the controller with the influence of the debris dynamics once grasped (Aghili, 2009a). This scheme is refined in the following to overcome this limitation.

Position/Speed/Acceleration Matching Eventually, the previous optimal problem of (Aghili, 2009a) is extended to reach the acceleration continuity at capture, searching for a smoother and safer capture. Indeed, this additional requirement must allow to track the



(a) (b)

Figure 7.3: Example of an optimal capture trajectory by matching position and speed; (a) X-axis components of the trajectory; (b) 3D capture trajectory.

grapple fixture for a short period of time, before locking the grasping mechanism of the end-effector of the chaser.

The third time derivative of position, or jerk, is then used as the new trajectory input:

$$\dddot{\mathbf{r}}_\mathrm{h} = \mathbf{u}_\mathrm{h} \qquad \text{and} \qquad \underbrace{\begin{bmatrix} \mathbf{r}_\mathrm{h}(t_f) - \mathbf{r}_\mathrm{t}(t_f) \\ \dot{\mathbf{r}}_\mathrm{h}(t_f) - \dot{\mathbf{r}}_\mathrm{t}(t_f) \\ \ddot{\mathbf{r}}_\mathrm{h}(t_f) - \ddot{\mathbf{r}}_\mathrm{t}(t_f) \end{bmatrix}}_{\psi(\boldsymbol{\chi}_\mathrm{h}(t_f))} = \mathbf{0}_{9\times 1} \tag{7.38}$$

The variables of the optimal problem are also re-defined as follows[2]:

$$\boldsymbol{\chi}_\mathrm{h} = \begin{bmatrix} \mathbf{r}_\mathrm{h}^\top & \boldsymbol{v}_\mathrm{h}^\top & \boldsymbol{a}_\mathrm{h}^\top \end{bmatrix}^\top \tag{7.39a}$$

$$\dot{\boldsymbol{\chi}}_\mathrm{h} = \begin{bmatrix} \boldsymbol{v}_\mathrm{h}^\top & \boldsymbol{a}_\mathrm{h}^\top & \mathbf{u}_\mathrm{h}^\top \end{bmatrix}^\top (= f_\mathrm{h}(\boldsymbol{\chi}_\mathrm{h}, \mathbf{u}_\mathrm{h})) \tag{7.39b}$$

$$L(\boldsymbol{\chi}_\mathrm{h}, \mathbf{u}_\mathrm{h}) = 1 + w_1 \|\boldsymbol{a}_\mathrm{h}\|^2 + w_u \|\mathbf{u}_\mathrm{h}\|^2 \tag{7.39c}$$

$$H(\boldsymbol{\chi}, \mathbf{u}_\mathrm{h}) = 1 + w_1 \|\boldsymbol{a}_\mathrm{h}\|^2 + w_u \|\mathbf{u}_\mathrm{h}\|^2 + \boldsymbol{\lambda}_\mathrm{d}^\top f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d}) + \boldsymbol{\lambda}_{\mathrm{h}_1}^\top \boldsymbol{v}_\mathrm{h} + \boldsymbol{\lambda}_{\mathrm{h}_2}^\top \boldsymbol{a}_\mathrm{h} + \boldsymbol{\lambda}_{\mathrm{h}_3}^\top \mathbf{u}_\mathrm{h} \tag{7.39d}$$

The problem is solved again with the same previous steps. $\mathbf{u}_\mathrm{h}$ is defined with (7.39b), and

[2]The weight on the speed is canceled in the cost function L, otherwise it leads to a third order ODE for $\mathbf{r}_\mathrm{h}$ that cannot be solved analytically. The difficult task in dealing with high order ODE is to be able to express the explicit dependency of the solution w.r.t. time, in order to use it in the implicit equation defining the final time.

by developing (7.14c), one obtains that:

$$\begin{aligned}\left(\frac{\partial H}{\partial \mathbf{u}}\right)^\top = \mathbf{0}_{3\times 1} &= \left(\frac{\partial f_\mathrm{h}}{\partial \mathbf{u}_\mathrm{h}}\right)^\top \boldsymbol{\lambda}_\mathrm{h} + \left(\frac{\partial L}{\partial \mathbf{u}_\mathrm{h}}\right)^\top \\ &= \left(\begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 \end{bmatrix}\right) \boldsymbol{\lambda}_\mathrm{h} + 2\, w_u\, \mathbf{u}_\mathrm{h} \\ &= \boldsymbol{\lambda}_{\mathrm{h}_3} + 2\, w_u\, \mathbf{u}_\mathrm{h}\end{aligned}$$

leading to:

$$\boldsymbol{\lambda}_{\mathrm{h}_3} = -2\, w_u\, \mathbf{u}_\mathrm{h} \tag{7.40}$$

The co-states dynamics are re-developed as:

$$\begin{aligned}\dot{\boldsymbol{\lambda}}_\mathrm{h} &= -\left(\frac{\partial f_\mathrm{h}}{\partial \boldsymbol{\chi}_\mathrm{h}}\right)^\top \boldsymbol{\lambda}_\mathrm{h} - \left(\frac{\partial L}{\partial \boldsymbol{\chi}_\mathrm{h}}\right)^\top \\ &= -\begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix}^\top \boldsymbol{\lambda}_\mathrm{h} - \begin{bmatrix} \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & 2w_1\, \boldsymbol{v}_\mathrm{h}^\top \end{bmatrix}^\top \\ &= \begin{bmatrix} \mathbf{0}_{3\times 1} \\ -\boldsymbol{\lambda}_{\mathrm{h}_1} \\ -\boldsymbol{\lambda}_{\mathrm{h}_2} - 2w_1\, \boldsymbol{v}_\mathrm{h} \end{bmatrix}\end{aligned}$$

Providing again that $\boldsymbol{\lambda}_{\mathrm{h}_1} = \boldsymbol{\lambda}_{\mathrm{h}_1}(0)$ is constant. Replacing $\mathbf{u}_\mathrm{h}$ by (7.38) in (7.40), the differential equation for $\mathbf{r}_\mathrm{h}$ reads as follows:

$$\frac{\mathrm{d}^4}{\mathrm{d}t^4}\left(\ddot{\mathbf{r}}_\mathrm{h} - \frac{w_1}{w_u}\mathbf{r}_\mathrm{h}\right) = \mathbf{0}_{3\times 1} \tag{7.41}$$

By still defining $\sigma = w_1/w_u > 0$, the previous ODE is integrated four times to obtain the RHS under a polynomial of higher order with $\alpha_0 + \alpha_1\, t + \alpha_2\, t^2 + \alpha_3\, t^3$. The homogeneous and particular solution are obtained with the same shapes, of an exponential homogeneous solution and a polynomial particular solution:

$$\begin{aligned}\mathbf{r}_\mathrm{h}^H &= \mathbf{k}_4\, e^{\sigma\, t} + \mathbf{k}_5\, e^{-\sigma\, t} \\ \mathbf{r}_\mathrm{h}^P &= \mathbf{k}_0 + \mathbf{k}_1\, t + \mathbf{k}_2\, t^2 + \mathbf{k}_3\, t^3\end{aligned}$$

leading to the following general solution:

$$\boxed{\mathbf{r}_\mathrm{h}(t) = \mathbf{k}_0 + \mathbf{k}_1\, t + \mathbf{k}_2\, t^2 + \mathbf{k}_3\, t^3 + \mathbf{k}_4\, e^{\sigma\, t} + \mathbf{k}_5\, e^{-\sigma\, t}} \tag{7.42}$$

Using the final condition in (7.38), **step 4** is defined by the following linear system:

$$\left[\begin{array}{cccc|cc} \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{I} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \sigma\,\mathbf{I}_3 & -\sigma\,\mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \sigma^2\,\mathbf{I}_3 & \sigma^2\,\mathbf{I}_3 \\ \hline \mathbf{I}_3 & t_f\mathbf{I}_3 & t_f^2\,\mathbf{I}_3 & t_f^3\,\mathbf{I}_3 & e^{\sigma\,t_f}\,\mathbf{I}_3 & e^{\sigma\,t_f}\,\mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & 2t_f\,\mathbf{I}_3 & 3t_f^2\,\mathbf{I}_3 & \sigma e^{\sigma\,t_f}\,\mathbf{I}_3 & -\sigma e^{\sigma\,t_f}\,\mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & 2\,\mathbf{I}_3 & 6t_f\,\mathbf{I}_3 & \sigma^2 e^{\sigma\,t_f}\,\mathbf{I}_3 & \sigma^2 e^{\sigma\,t_f}\,\mathbf{I}_3 \end{array}\right] \left[\begin{array}{c} \mathbf{k}_0 \\ \mathbf{k}_1 \\ \mathbf{k}_2 \\ \mathbf{k}_3 \\ \mathbf{k}_4 \\ \mathbf{k}_5 \end{array}\right] = \left[\begin{array}{c} \mathbf{r}_\mathrm{h}(0) \\ \boldsymbol{v}_\mathrm{h}(0) \\ \boldsymbol{a}_\mathrm{h}(0) \\ \hline \mathbf{r}_\mathrm{t}(t_f) \\ \dot{\mathbf{r}}_\mathrm{t}(t_f) \\ \ddot{\mathbf{r}}_\mathrm{t}(t_f) \end{array}\right] \tag{7.43}$$

The same previous alternative can be developed, without the weight one the acceleration in L. The ODE becomes:

$$\frac{\mathrm{d}^4}{\mathrm{d}t^4}\left(\ddot{\mathbf{r}}_\mathrm{h}\right) = \mathbf{0}_{3\times1} \tag{7.44}$$

and lead to the following polynomial solution:

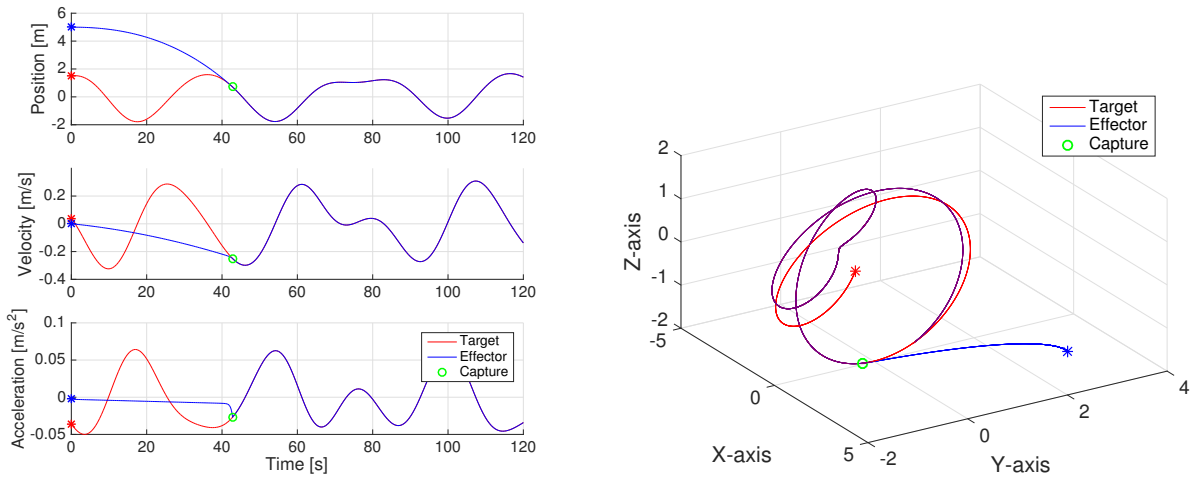
$$\mathbf{r}_\mathrm{h}(t) = \mathbf{k}_0 + \mathbf{k}_1\,t + \mathbf{k}_2\,t^2 + \mathbf{k}_3\,t^3 + \mathbf{k}_4\,t^4 + \mathbf{k}_5\,t^5 \tag{7.45}$$

The new linear system reads as follows:

$$\left[\begin{array}{ccc|ccc} \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \hline \mathbf{I}_3 & t_f\,\mathbf{I}_3 & t_f^2\,\mathbf{I}_3 & t_f^3\,\mathbf{I}_3 & t_f^4\,\mathbf{I}_3 & t_f^5\,\mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & 2t_f\,\mathbf{I}_3 & 3t_f^2\,\mathbf{I}_3 & 4t_f^3\,\mathbf{I}_3 & 5t_f^4\,\mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & 2\mathbf{I}_3 & 6t_f\,\mathbf{I}_3 & 12t_f^2\,\mathbf{I}_3 & 20t_f^3\,\mathbf{I}_3 \end{array}\right] \left[\begin{array}{c} \mathbf{k}_0 \\ \mathbf{k}_1 \\ \mathbf{k}_2 \\ \mathbf{k}_3 \\ \mathbf{k}_4 \\ \mathbf{k}_5 \end{array}\right] = \left[\begin{array}{c} \mathbf{r}_\mathrm{h}(0) \\ \boldsymbol{v}_\mathrm{h}(0) \\ \boldsymbol{a}_\mathrm{h}(0) \\ \hline \mathbf{r}_\mathrm{t}(t_f) \\ \dot{\mathbf{r}}_\mathrm{t}(t_f) \\ \ddot{\mathbf{r}}_\mathrm{t}(t_f) \end{array}\right] \tag{7.46}$$

Eventually, the implicit equation in the final time is given as follows to complete **step 5**:

$$\begin{aligned} H(t_f) = 0 &= \left(1 + w_1\,\|\boldsymbol{a}_\mathrm{h}\|^2 + w_u\,\mathbf{u}_\mathrm{h}(t_f)^\top\mathbf{u}_\mathrm{h}(t_f) + \boldsymbol{\lambda}_\mathrm{d}(t_f)^\top f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d}) + \boldsymbol{\lambda}_{\mathrm{h}_1}^\top\,\boldsymbol{v}_\mathrm{h} + \boldsymbol{\lambda}_{\mathrm{h}_2}^\top\,\boldsymbol{a}_\mathrm{h} + \boldsymbol{\lambda}_{\mathrm{h}_3}^\top\,\mathbf{u}_\mathrm{h}\right)_{t_f} \\ &= \left(1 + w_1\,\ddot{\mathbf{r}}_\mathrm{h}^\top\ddot{\mathbf{r}}_\mathrm{h} + w_u\,\dddot{\mathbf{r}}_\mathrm{h}^\top\dddot{\mathbf{r}}_\mathrm{h} + \left(\boldsymbol{\lambda}_\mathrm{d}^\top\,f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d})\right)_{t_f} + \boldsymbol{\lambda}_{\mathrm{h}_1}^\top\,\dot{\mathbf{r}}_\mathrm{h} + \boldsymbol{\lambda}_{\mathrm{h}_2}^\top\,\ddot{\mathbf{r}}_\mathrm{h} - 2w_u\,\dddot{\mathbf{r}}_\mathrm{h}^\top\,\dddot{\mathbf{r}}_\mathrm{h}\right)_{t_f} \\ &= 1 + w_1\,\ddot{\mathbf{r}}_\mathrm{h}(t_f)^\top\ddot{\mathbf{r}}_\mathrm{h}(t_f) - w_u\,\dddot{\mathbf{r}}_\mathrm{h}(t_f)^\top\dddot{\mathbf{r}}_\mathrm{h}(t_f) + \boldsymbol{\lambda}_\mathrm{d}(t_f)^\top\,f_\mathrm{d}(\boldsymbol{\chi}_\mathrm{d}) + \boldsymbol{\lambda}_{\mathrm{h}_1}^\top\,\dot{\mathbf{r}}_\mathrm{h}(t_f) + \boldsymbol{\lambda}_{\mathrm{h}_2}^\top\,\ddot{\mathbf{r}}_\mathrm{h}(t_f) \end{aligned}$$



(a) (b)

Figure 7.4: Example of an optimal capture trajectory by matching position, speed and acceleration; (a) X-axis components of the trajectory; (b) 3D capture trajectory.

The unknown $\boldsymbol{\lambda}_{\mathrm{d}}(t_f)$ is still obtained using (7.13e):

$$\begin{bmatrix} \boldsymbol{\lambda}_{\mathrm{d}} \\ \boldsymbol{\lambda}_{\mathrm{h}} \end{bmatrix}_{t_f} = \left(\nu_{\psi}^{\top} \begin{bmatrix} \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{\chi}_{\mathrm{d}}} & \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{\chi}_{\mathrm{h}}} \end{bmatrix}\right)_{t_f}^{\top} = \begin{bmatrix} -\left(\frac{\partial \mathbf{r}_{\mathrm{t}}}{\partial \boldsymbol{\chi}_{\mathrm{d}}}\right)^{\top} \nu_{\psi_1} - \left(\frac{\partial \dot{\mathbf{r}}_{\mathrm{t}}}{\partial \boldsymbol{\chi}_{\mathrm{d}}}\right)^{\top} \nu_{\psi_2} \\ -\left(\frac{\partial \ddot{\mathbf{r}}_{\mathrm{t}}}{\partial \boldsymbol{\chi}_{\mathrm{d}}}\right)^{\top} \nu_{\psi_3} \\ \nu_{\psi_1} \\ \nu_{\psi_2} \\ \nu_{\psi_3} \end{bmatrix}_{t_f}$$

which leads to $\nu_{\psi_1} = \boldsymbol{\lambda}_{\mathrm{h}_1}(t_f)$, $\nu_{\psi_2} = \boldsymbol{\lambda}_{\mathrm{h}_2}(t_f)$ and $\nu_{\psi_3} = \boldsymbol{\lambda}_{\mathrm{h}_3}(t_f)$. Using (7.40), the last co-states of the debris dynamics are eliminated by:

$$\begin{aligned} \boldsymbol{\lambda}_{\mathrm{d}}(t_f)^{\top} f_{\mathrm{d}}(\boldsymbol{\chi}_{\mathrm{d}}) &= \left(\boldsymbol{\lambda}_{\mathrm{h}_1}(t_f)^{\top} \left(-\frac{\partial \mathbf{r}_{\mathrm{t}}}{\partial \boldsymbol{\chi}_{\mathrm{d}}}\right) + \boldsymbol{\lambda}_{\mathrm{h}_2}(t_f)^{\top} \left(-\frac{\partial \dot{\mathbf{r}}_{\mathrm{t}}}{\partial \boldsymbol{\chi}_{\mathrm{d}}}\right) + \boldsymbol{\lambda}_{\mathrm{h}_3}(t_f)^{\top} \left(-\frac{\partial \ddot{\mathbf{r}}_{\mathrm{t}}}{\partial \boldsymbol{\chi}_{\mathrm{d}}}\right)\right) f_{\mathrm{d}}(\boldsymbol{\chi}_{\mathrm{d}}) \\ &= -\boldsymbol{\lambda}_{\mathrm{h}_1}(t_f)^{\top} \dot{\mathbf{r}}_{\mathrm{t}}(t_f) - \boldsymbol{\lambda}_{\mathrm{h}_2}(t_f)^{\top} \ddot{\mathbf{r}}_{\mathrm{t}}(t_f) + 2w_u \dddot{\mathbf{r}}_{\mathrm{h}}(t_f)^{\top} \dddot{\mathbf{r}}_{\mathrm{t}}(t_f) \end{aligned}$$

Eventually, the final conditions on speed and acceleration in (7.38) brings out the last simplifications in the implicit equation defining t_f. This latter reads as follows:

$$\boxed{1 + w_1 \ddot{\mathbf{r}}_{\mathrm{h}}(t_f)^{\top} \ddot{\mathbf{r}}_{\mathrm{h}}(t_f) - w_u \dddot{\mathbf{r}}_{\mathrm{h}}(t_f)^{\top} \dddot{\mathbf{r}}_{\mathrm{h}}(t_f) + 2w_u \dddot{\mathbf{r}}_{\mathrm{h}}(t_f)^{\top} \dddot{\mathbf{r}}_{\mathrm{t}}(t_f) = 0} \tag{7.47}$$

The results providing the continuity in acceleration are illustrated in Figure 7.4 for the initial condition in (7.23).

Conclusion

This last guidance scheme is more suited to a smooth capture since it allows to track the grapple fixture for an unlimited period of time. With this approach, the end-effector of the chaser is first synchronized with the target natural motion, and then the locking mechanism can capture the debris without impact. Even if the grasping should not be too long in a real mission, this smoother transition at the guidance level allow to better compensate for potential errors on the debris dynamics propagation.

A more general optimal problem is also proposed in (Aghili, 2012), to include the error measurement between the end-effector and the target point into the optimal planner. It allows to continuously update the capture trajectory while the debris could be disturbed during the period $t \in [0, t_f]$. Another extension is proposed in (Aghili, 2009b) with the consideration of the post-capture stabilization for a tumbling debris. The angular momentum brought by the debris at the instant of capture is dissipated by a time optimal control minimizing the duration of the stabilization maneuver. As mentioned above, an instantaneous transition is assumed between the controller of the free manipulator, and the loaded one at the instant capture, which seems quite optimistic.

CHAPTER 8 SPACE ROBOT CONTROL

The control structure of the space robot aims at finding the best compromise between the computational complexity and the performance requirements to capture a slow-tumbling debris. To that end, the controller is split into two loops. One low-level loop is merging the manipulator joint control and the spacecraft AOCS, while the upper level is almost a guidance loop to ensure that the end-effector is tracking the target. In the following, the control architecture is first described, then the controller synthesis scheme is introduced to fulfill the mission requirements. Finally, the last section is dedicated to the robust analysis of the robotic controller through a novel LFT modeling of the system mass matrix.

8.1 Control Architecture

Two different loops are used to control the robot efficiently. In the first one, the system measurements of the base and of the manipulator are centralized to perform high-level tasks, like a trajectory tracking by the base or by the end-effector. Among these schemes, the *free-flying* and *free-floating* cases were mentioned in the literature review. The first one lies on an active control of the base, while the second one lets the base move freely in reaction to the arm motion (Dubowsky and Papadopoulos, 1993; Moosavian and Papadopoulos, 2007a). It is even suggested by E. Papadopoulos and S. Dubowsky that any classic terrestrial control scheme may be applied to the space robots for a limited workspace, provided that dynamic singularities are avoided (Papadopoulos and Dubowsky, 1991a).

In the sequel, the *free-flying* mode is preferred since the base attitude must be maintained at almost any time during the arm motion. This is a strong requirement if the space robot is tele-operated from the ground and if antennas need to be accurately pointed. As proposed by M. Oda in (Oda, 1994), the global controller of the multi-body system is split into two distinct structures: a traditional AOCS for the base, commonly tackled by space system engineers, and a manipulator controller, designed by robotics engineers. The main drawback of this decoupling is the potential saturation of the AOCS if the arm moves too fast. A feedforward controller could be proposed to improve the global performance by roughly estimating the disturbance induced by the arm (Oda, 1996). As an alternative, a complete coordinated control is available in (Papadopoulos and Dubowsky, 1991b) through an adapted Inverse Jacobian control, to achieve a position and orientation control of both the end-effector and the base.

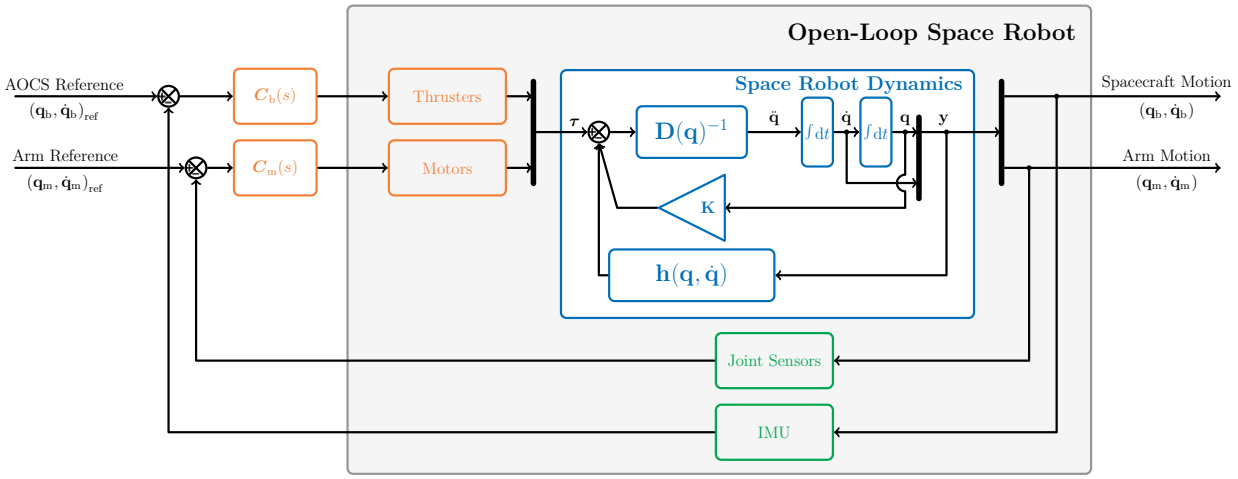


Figure 8.1: Decentralized control architecture between the AOCS and the arm's controller

One major limitation of space systems is coming from the processors available for space missions, which are significantly less powerful than the terrestrial ones due to their higher robustness. A low complexity control architecture will be preferred in this study to anticipate the limited processing capability of on-board processors. Therefore, the control architecture presented here lies on the design of mere PD controllers at the joints and at the base. The coupling between both structures is accounted for during the synthesis, the aim being to obtain the highest performances possible with such simple architecture.

8.1.1 Inner Loop

The system inner loop is illustrated in Figure 8.1. The measurements for the spacecraft are assumed to come from an IMU and a *Global Positioning System (GPS)* providing both the inertial velocities and the inertial position/attitude (Weeden, 2011). Regarding the actuators, only the thrusters are considered here since the manipulator motion is implying efforts much higher than the usual orbital disturbances, as it will be seen in the following. At the arm level, the measurements are embedded in the joint themselves and provide the position and velocity of the link along its axis of motion. Of course, the links are actuated by the joints at their base.

Although the actuators were considered perfect in this thesis, more accurate models of the latter could easily be included in future works, thanks to the thorough derivations of multi-

body system dynamics. For example, it was mentioned that the reaction wheels could be considered as additional appendages driven by control torques. Doing this, their dynamic coupling with the base would be represented more precisely. In a similar fashion, the joint actuators could be refined by accounting for the shaft flexibility with the addition of a cylinder link between the end of a segment and the beginning of the following one. The torsion of this body may account faithfully for the real deflection due to the joint elasticity.

For the spacecraft, a PD controller is used with a decoupled behavior w.r.t. the linear and angular dynamics. Its structure is chosen as a compromise between a full order controller, that brings better performances but at the expense of a higher computational burden, and a decoupled controller, that limits performances but remains simple enough to be compatible with the processors capabilities. The complete decoupling on each axis could also be considered, but the angular dynamics are kept coupled because the arm is introducing high asymmetric inertias that prevent this decoupling. The AOCS controller takes the following form, where the "*pos*" stands for the control in position, and "*rot*" for the one in rotation.

$$\boldsymbol{C}_{\mathrm{b}} = \left[\begin{array}{cc|cc} \boldsymbol{K}_{p_{pos}} & \mathbf{0}_{3\times3} & \boldsymbol{K}_{d_{pos}} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{K}_{p_{rot}} & \mathbf{0}_{3\times3} & \boldsymbol{K}_{d_{rot}} \end{array}\right] \tag{8.1}$$

For the arm, two structures are investigated: a centralized controller gathers all the joint measurements before specifying the joints command, while a decentralized one gives the torque commands of each joint by only using its local measurement. These two cases are studied to emphasize the best compromise in terms of closed-loop performances and computational burden. It will be shown how a decentralized controller yields convincing results when it is well tuned through the sophisticated structured H_∞ method.

Centralized joint control For the centralized architecture, the arm controller includes a nominal mass matrix $\mathbf{D}_0$ obtained for a given configuration $\mathbf{q}_0$ along the trajectory. This term reproduces the same coupling as in the nonlinear dynamics, and is clearly derived from the feedback linearization technique. A classic feedback linearization would update the mass matrix expression $\mathbf{D}(\mathbf{q})$ along the trajectory. In order to reduce on-line computation, a mean configuration $\mathbf{q}_0$ is used to compute the mass matrix only once and to provide partial information about the dynamic coupling. Obviously this structure is valid for a reduced set of joint motions, since the mass matrix terms are highly nonlinear and may vary greatly depending on the joint configuration. Merging the joint positions and velocities in a single vector, the centralized controller for the manipulator takes the form:

$$\boldsymbol{C}_{cen} = \mathbf{D}(\mathbf{q}_0) \left[\begin{array}{cccc|cccc} \boldsymbol{K}_{p_1} & 0 & \dots & 0 & \boldsymbol{K}_{d_1} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \boldsymbol{K}_{p_n} & 0 & \dots & 0 & \boldsymbol{K}_{d_n} \end{array}\right] \tag{8.2}$$

It is worth noticing that this controller is not defined by a full gain matrix, but by a set of design parameters instead. These gains $(\boldsymbol{K}_{p_i}, \boldsymbol{K}_{d_i})$ are local to the joints, such that the centralized feature of the controller lies in its dynamic coupling through the term $\mathbf{D}(\mathbf{q}_0)$.

Decentralized joint control For the decentralized architecture, the joint PD controllers are simultaneously synthesized considering the whole arm dynamics. With this architecture, each of them only uses its local measurements of position and velocity. The coupling between the joints and the base, or between the joint themselves, is only considered through the synthesis since the whole model is coupled. This controller is the most simple one can think of, but it will appear that the performance gap between both structures is not so important due to the slow dynamics involved when capturing a target tumbling at a modest rate. The decentralized controller is given as follows:

$$\boldsymbol{C}_{dec} = \left[\begin{array}{cccc|cccc} \boldsymbol{K}_{p_1} & 0 & \dots & 0 & \boldsymbol{K}_{d_1} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \boldsymbol{K}_{p_n} & 0 & \dots & 0 & \boldsymbol{K}_{d_n} \end{array}\right] \tag{8.3}$$

8.1.2 Outer Loop

The outer loop consists in the synchronization of the spacecraft and the manipulator controllers in order to achieve the target tracking by the end-effector. Called *coordinated control* in (Papadopoulos and Dubowsky, 1991b), this control is based on the global Jacobian matrix presented in (6.25). Considering a single manipulator mounted on a spacecraft with other appendages, like solar panels or antennas, this relation can be re-written in a simpler form by denoting $\mathbf{J}$ the Jacobian of this unique end-effector:

$$\mathbf{t}_{\mathrm{E}} = \mathbf{J}_{\mathrm{b}}\, \dot{\mathbf{q}}_{\mathrm{b}} + \mathbf{J}_{\mathrm{m}}\, \dot{\mathbf{q}}_{\mathrm{m}} \tag{8.4}$$

with $(\mathbf{q}_{\mathrm{b}}, \mathbf{q}_{\mathrm{m}})$ the states of the spacecraft and of the arm, and $\mathbf{J}_{\mathrm{b}} \in \mathbb{R}^{6\times 6}$ the rigid motion influence of the spacecraft, and $\mathbf{J}_{\mathrm{m}} \in \mathbb{R}^{6\times n}$ the traditional Jacobian matrix, already mentioned in (4.33).

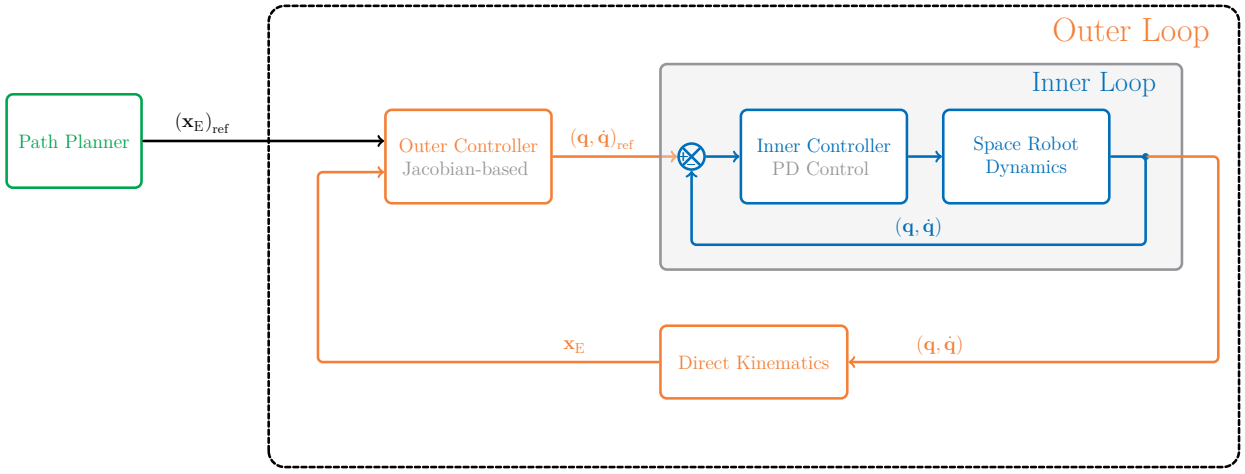


Figure 8.2: Outer control loop of the space robot based on the Jacobian matrix

The global structure of the outer loop is illustrated in Figure 8.2. The strategy adopted here is to set the spacecraft position at a safety distance from the debris, and to move the robotic arm to compensate for the base motion and track the target in any case. To this end, the manipulability index of the Jacobian matrix is computed for the motion of the three main segments. It appears to be almost invariant along the first joint due to the symmetric shape of the base. Hence, the index is plotted in Figure 8.3 when the arm is moving in a vertical plane (X, Y), orthogonal to the manipulator base. For each position the inverse kinematics is reduced to the computation of the angles θ_2 and θ_3, and the corresponding manipulability index is computed with (4.35), using the global Jacobian matrix in (6.26) as proposed in (Umetani and Yoshida, 2001).

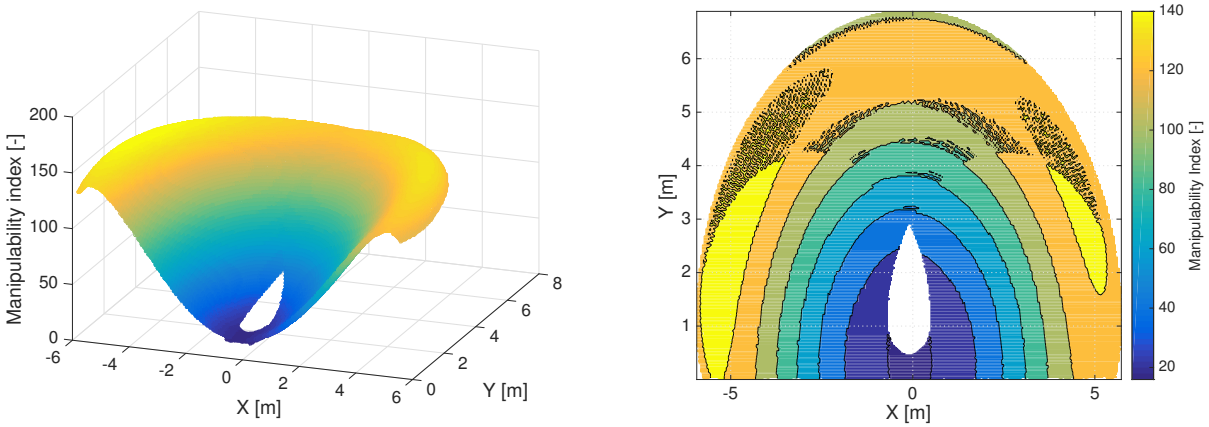
According to this map, the manipulability index is maximized in an area almost circular from 4 to 5 meters from the manipulator anchorage point. This area of maximum manipulability is denoted by $\mathcal{A}_m$ in the following, and is defined by the sphere of radius $d_{\mathcal{A}} = 4.5m$. The strategy of guidance reads:

Step 1 Compute the distance of the desired effector trajectory from the manipulator anchorage point;

Step 2 Move the spacecraft toward a position $\mathbf{r}_{b_{ref}}$ that ensures the end-effector lies inside $\mathcal{A}_m$;

Step 3 Using the space robot Jacobian matrix, compute the resulting manipulator configuration $\mathbf{q}_{m_{ref}}$ that ensures the desired position to be reached.

At the mathematical level, this guidance strategy lies on the Jacobian matrix decomposition proposed in (8.4). The differential motion required on the base aims at moving it to bring the desired effector position closer to $\mathcal{A}_m$, while the differential motion of the arm compensate



(a) (b)

Figure 8.3: Manipulability index of the manipulator of the space robot in a vertical plane around its base

this base behavior to reach the desired effector position in any case. Denoting by $d_{\mathrm{m,E_{ref}}}$ the distance between the arm base and the desired effector position, i.e., the norm of the vector $\mathbf{r}_{\mathrm{m,E_{ref}}}$, the incremental motion of the base and of the arm are given as follows:

$$\delta\mathbf{r}_{\mathrm{b}} = \left(d_{\mathrm{m,E_{ref}}} - d_{\mathcal{A}}\right) \frac{\mathbf{r}_{\mathrm{m,E_{ref}}}}{d_{\mathrm{m,E_{ref}}}}$$

$$\delta\mathcal{Q}_{\mathrm{b}} = \tilde{\mathcal{Q}}_{\mathrm{b}} \otimes \mathcal{Q}_{\mathrm{b_{ref}}}$$

$$\delta\mathbf{q}_{\mathrm{m}} = \mathbf{J}_{\mathrm{m}}^{-1} \left(\begin{bmatrix} \mathbf{r}_{\mathrm{E_{ref}}} - \mathbf{r}_{\mathrm{E}} \\ \tilde{\mathcal{Q}}_{\mathrm{E}} \otimes \mathcal{Q}_{\mathrm{E_{ref}}} \end{bmatrix} - \mathbf{J}_{\mathrm{b}} \begin{bmatrix} \delta\mathbf{r}_{\mathrm{b}} \\ \delta\mathcal{Q}_{\mathrm{b}} \end{bmatrix} \right)$$

where $\tilde{\mathcal{Q}}$ denotes the inverse of a quaternion, and $\mathcal{Q}_{\mathrm{b_{ref}}}$ is an attitude of reference for the base, which is assumed to be provided.

With these kinematic relations, the desired position and orientation of the base and the desired configuration of the manipulator are given by :

$$\mathbf{r}_{\mathrm{b_{ref}}} = \mathbf{r}_{\mathrm{b}} + \delta\mathbf{r}_{\mathrm{b}} \left(\mathbf{r}_{\mathrm{E_{ref}}}\right)$$

$$\mathbf{q}_{\mathrm{m_{ref}}} = \mathbf{q}_{\mathrm{m}} + \delta\mathbf{q}_{\mathrm{m}} \left(\mathbf{x}_{\mathrm{b}}, \mathbf{q}_{\mathrm{m}}, \mathbf{x}_{\mathrm{E_{ref}}}\right)$$

merging the position and the quaternion of the base and the desired effector motion into the states $\mathbf{x}_{\mathrm{b}}$ and $\mathbf{x}_{\mathrm{E_{ref}}}$.

Considering that these incremental motions are small if the end-effector is following accurately the desired trajectory, they are considered as the velocity requirements for the inner loop, while the base and the arm coordinates are obtained by integrating them over time. To

that end, the gains (G_b, G_m) are introduced to tune the guidance loop, and include the sampling time of the computations, in order to respect the velocity units. The global vectors of reference sent to the inner loop read as follows:

$$\dot{\mathbf{q}}_{\mathrm{b}_{\mathrm{ref}}} = G_\mathrm{b} \begin{bmatrix} \delta\mathbf{r}_\mathrm{b} \\ \delta\mathcal{Q}_\mathrm{b}(2{:}4) \end{bmatrix} \quad \text{and} \quad \dot{\mathbf{q}}_{\mathrm{m}_{\mathrm{ref}}} = G_\mathrm{m}\, \delta\mathbf{q}_\mathrm{m} \tag{8.5a}$$

$$\mathbf{q}_{\mathrm{b}_{\mathrm{ref}}} = \int \dot{\mathbf{q}}_{\mathrm{b}_{\mathrm{ref}}}\, \mathrm{d}t \quad \text{and} \quad \mathbf{q}_{\mathrm{m}_{\mathrm{ref}}} = \int \dot{\mathbf{q}}_{\mathrm{m}_{\mathrm{ref}}}\, \mathrm{d}t \tag{8.5b}$$

8.2 Controller Synthesis

For both control strategies at the manipulator level, the synthesis is performed following the same steps. As explained in (Kawamura et al., 1988), trajectory tracking may be efficiently performed with this type of controller, if the derivative action is sufficiently high. Nevertheless, it cannot be increased indefinitely in practice since the control torques are limited at the joints or by the thrusters capabilities. Hence, gain limits are fixed during the synthesis to reach the best compromise of performance while meeting the actuators requirements.

8.2.1 List of requirements

The control requirements for the space robot are summarized in Table 8.1. They give the actuators capabilities and the expected performances on the base attitude and on the end-effector tracking errors.

Based on the ETS-VII mission (Oda, 1994, 1997), the base attitude error must be maintained under $\mathbf{0.3}^\circ$- $\mathbf{0.5}^\circ$ when the arm is fixed, and under $\mathbf{3}^\circ$ when it is moving. These requirements are used to maintain a pointing accuracy toward the Earth for communication purpose during the capture, and also to ensure the solar panels exposition to the Sun for power generation. At the actuator level, the thruster torques are limited to $\mathbf{5\,Nm}$ considering that the spacecraft is only a few meters long, and that the nominal force of a thruster is 20 N in (Oda, 1994). In comparison, the maximum reaction wheels capabilities were 0.1 Nm on the ETS-VII mission.

Considering the robotic arm, the end-effector must accurately track the circular path used as a capture trajectory. The tumbling rate is set to $\boldsymbol{\omega}_\mathrm{t} = \mathbf{4}^\circ/\mathbf{s}$ according to (Lampariello, 2010). For the capture phase, taking the case of the *SARAH* hand, developed by the Canadian Space Agency and Laval University (Rubinger et al., 2002), the end-effector has a grasping capability of 10 cm for a cylinder shaped object. The tracking error is thus required to be

below the tenth of this value with **1 cm**, in order to prevent any slip at the instant of capture. Regarding the actuators, the maximum joint torques are given by the MDA design provided to TAS. The arm is given into details in Appendix D.1.3. It weighs 63 kg, and the joint torques are limited to **200 Nm**. As a comparison, the Canadarm weighs 450 kg and the joints can develop 1500 Nm.

Table 8.1: List of constraints and requirements

Base		Arm	
Torques	Attitude	Torques	Tip position
±5 Nm	**±0.3°**	**±200 Nm**	**±1 cm**

8.2.2 Linearized model

Since the synthesis is based on linear techniques like the H_∞ method, the nonlinear dynamics of the space system must be linearized to obtain the control models. Coming back to the expression of the space robot dynamics in (6.47), one may notice that the nonlinear terms are still quadratic in the generalized velocities, as it was the case for the fixed-base manipulator (Spong et al., 2006). This result allows to greatly simplify the nonlinear model. Considering an equilibrium position of the system $\mathbf{q}_0$ without external efforts, the state equations are given by:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\mathbf{q}_0\\ \dot{\mathbf{q}}_0\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\dot{\mathbf{q}}_0\\ \mathbf{D}(\mathbf{q}_0)^{-1}\left(\boldsymbol{\tau}_0 - \mathbf{h}(\mathbf{q}_0,\,\dot{\mathbf{q}}_0) - \mathbf{K}\,\mathbf{q}_0\right)\end{bmatrix}$$

One obtains that the generalized velocities are $\mathbf{0}$ at the equilibrium, implying that the Coriolis and centrifugal vector vanishes. The generalized efforts $\boldsymbol{\tau}$ and the stiffness terms $\mathbf{K}\,\mathbf{q}$ apply on different sets of equations and do not overlap one with another. Indeed, the dynamic equation could be re-written in a decoupled form as:

$$\begin{aligned}\ddot{\mathbf{q}}_{\mathrm{r},0} &= (\mathbf{D}^{-1})_{\mathrm{rr}}\,\boldsymbol{\tau}_{\mathrm{r}}\\ \ddot{\mathbf{q}}_{\mathrm{f},0} &= (\mathbf{D}^{-1})_{\mathrm{fr}}\,\boldsymbol{\tau}_{\mathrm{r}} - (\mathbf{D}^{-1})_{\mathrm{ff}}\,\mathbf{K}_{\mathrm{ff}}\,\mathbf{q}_{\mathrm{f}_0}\end{aligned}$$

where the inverse of the mass matrix, the global stiffness matrix, and the generalized efforts are partitioned according to the rigid and flexible coordinates. It must be recalled from (5.62) that the stiffness matrix exhibits zero sub-blocks w.r.t. the rigid coordinates. These equations provide that the generalized efforts are $\mathbf{0}$ at the equilibrium too, since the mass matrix is always positive-definite and thus invertible. A secondary implication of the equilibrium is that the deformation described by the flexible coordinates $\mathbf{q}_{\mathrm{f}_0}$ must be $\mathbf{0}$. Therefore,

the free-floating case with undeformed segments and without joint torques is an equilibrium position for the space robot, whatever the configuration. This conclusion differs from the Earth-based manipulators, for which a given torque must be applied at the joints to maintain an equilibrium and compensate for the gravity.

Around a given nominal static configuration of the system $\mathbf{q}_0$, the variations around this equilibrium are denoted by:

$$\delta\mathbf{q} = \left[\begin{array}{ccc} \delta\mathbf{r}_{\mathrm{b}}^{\top} & \delta\boldsymbol{\Psi}_{\mathrm{b}}^{\top} & \delta\mathbf{q}_{\mathrm{a}}^{\top} \end{array} \right]^{\top} \tag{8.6}$$

where $\mathbf{q}_{\mathrm{a}}$ stands for the appendages coordinates[1]. The Euler angles are used to denote the attitude motion of the base, but a quaternion could also be considered by augmenting accordingly the size of the vector $\mathbf{q}$ and of the matrix $\mathbf{K}$.

The differential model around this equilibrium is obtained as follows:

$$\mathbf{D}(\mathbf{q}_0 + \delta\mathbf{q})\ (\ddot{\mathbf{q}}_0 + \delta\ddot{\mathbf{q}}) + \mathbf{h}\left(\mathbf{q}_0 + \delta\mathbf{q}, \dot{\mathbf{q}}_0 + \delta\dot{\mathbf{q}}\right) + \mathbf{K}\ (\mathbf{q}_0 + \delta\mathbf{q}) = \boldsymbol{\tau}_0 + \delta\boldsymbol{\tau}$$

Applying the previous results of the equilibrium, i.e., $\mathbf{q}_{\mathrm{f}_0} = \mathbf{0}$ and $\dot{\mathbf{q}}_0 = \ddot{\mathbf{q}}_0 = \boldsymbol{\tau}_0 = \mathbf{0}$, the dynamic equation leads to:

$$\mathbf{D}(\mathbf{q}_0 + \delta\mathbf{q})\,\delta\ddot{\mathbf{q}} + \mathbf{h}\left(\mathbf{q}_0 + \delta\mathbf{q}, \delta\dot{\mathbf{q}}\right) + \mathbf{K}\,\delta\mathbf{q} = \delta\boldsymbol{\tau}$$

where the partitioning of the stiffness matrix provides that: $\mathbf{q}_{\mathrm{f}_0} = \mathbf{0} \implies \mathbf{K}\,\mathbf{q}_0 = \mathbf{0}$. Besides, recalling that the Coriolis and centrifugal terms are quadratic in the velocities, the linear model is finally extracted under the classic following mechanical model:

$$\mathbf{D}(\mathbf{q}_0)\,\delta\ddot{\mathbf{q}} + \mathbf{K}\,\delta\mathbf{q} = \delta\boldsymbol{\tau} \tag{8.7}$$

Using the control notations, one obtains the synthesis model as :

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\begin{array}{c} \mathbf{x} \\ \dot{\mathbf{x}} \end{array} \right] = \left[\begin{array}{cc} \mathbf{0}_{N\times N} & \mathbf{I}_N \\ -\mathbf{D}(\mathbf{q}_0)^{-1}\,\mathbf{K} & \mathbf{0}_{N\times N} \end{array} \right] \left[\begin{array}{c} \mathbf{x} \\ \dot{\mathbf{x}} \end{array} \right] + \left[\begin{array}{c} \mathbf{0}_{N\times N} \\ \mathbf{D}(\mathbf{q}_0)^{-1} \end{array} \right] \boldsymbol{\tau} \tag{8.8}$$

with the states given by: $\mathbf{x} = \delta\mathbf{q} \in \mathbb{R}^N$. If damping terms were considered on the flexible modes, the lower right-hand sub-matrix would be replaced by $-\mathbf{D}(\mathbf{q}_0)^{-1}\,\mathbf{B}$, with $\mathbf{B}$ the damping matrix. In addition, stiffness and damping terms could also be added at the joint

[1] In the case of a space robot with a single robotic arm described by $\mathbf{q}_{\mathrm{m}}$, this vector could contain the additional coordinates of flexible solar panels for example.

level, and would appear in the rigid dynamics. The resulting matrices $\mathbf{K}$ and $\mathbf{B}$ would stay decoupled between the two types of flexibility, i.e., at the joints and along the segments.

The final input/output model of the linearized dynamics is given by:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix} = \begin{bmatrix}\mathbf{0}_{N\times N} & \mathbf{I}_N\\ -\mathbf{D}(\mathbf{q}_0)^{-1}\mathbf{K} & \mathbf{0}_{N\times N}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix} + \begin{bmatrix}\mathbf{0}_{N\times N} & \mathbf{0}_{N\times N}\\ \mathbf{I}_N & \mathbf{D}(\mathbf{q}_0)^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{w}\\ \boldsymbol{\tau}\end{bmatrix} \tag{8.9}$$

$$\begin{bmatrix}\mathbf{a}\\ \mathbf{y}\end{bmatrix} = \left[\begin{array}{cc}-\mathbf{D}(\mathbf{q}_0)^{-1}\mathbf{K} & \mathbf{0}_{N\times N}\\ \hline \mathbf{I}_N & \mathbf{0}_{N\times N}\\ \mathbf{0}_{N\times N} & \mathbf{I}_N\end{array}\right]\begin{bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{bmatrix} + \left[\begin{array}{cc}\mathbf{I}_N & \mathbf{D}(\mathbf{q}_0)^{-1}\\ \hline \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N}\\ \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N}\end{array}\right]\begin{bmatrix}\mathbf{w}\\ \boldsymbol{\tau}\end{bmatrix} \tag{8.10}$$

The complete model used for the controller synthesis is illustrated in Figure 8.4. A disturbance signal is added on the acceleration, and will be considered as an exogenous input in the H_∞ design presented in the next section. To that end, this transfer on the acceleration is weighted by $\mathbf{W}(s)$ in order to reject the disturbances.

8.2.3 Synthesis scheme

Once the controller structure is fixed, stability and performances are ensured by carefully tuning its gains. As highlighted by (Kawamura et al., 1988), a PD controller achieves any trajectory tracking performance with a high derivative action. However, limited gain values are always considered to prevent the actuators saturation, while a trajectory tracking performance is still desired. With such constraints, controller gains cannot be tuned empirically.

The previous linearized model in (8.9) is used to tune the gains of the AOCS in (8.1) and of the arm in (8.2) and (8.3). The H_∞ framework is used by specifying frequency requirements on the closed-loop system using weighting functions on the desired transfers (Alazard et al., 1999; Skogestad and Postlethwaite, 2005). The synthesis is performed using the *Robust Control Toolbox* of `Matlab`, with the advanced functionalities of the *structured* H_∞ synthesis. Indeed, the function `systune` allows to define structured controller such as the decoupling forms introduced on the spacecraft and on the arm (Apkarian, 2013).

Weightings The transfer on the acceleration sensitivity function is used to improve the disturbance rejection in priority (Alazard, 2013). The following weighting function $\mathbf{W}(s)$ enables to reject efficiently any disturbance like the unmodeled dynamics of the gravity gradient, of the atmospheric drag or of the solar pressure that were neglected above. The

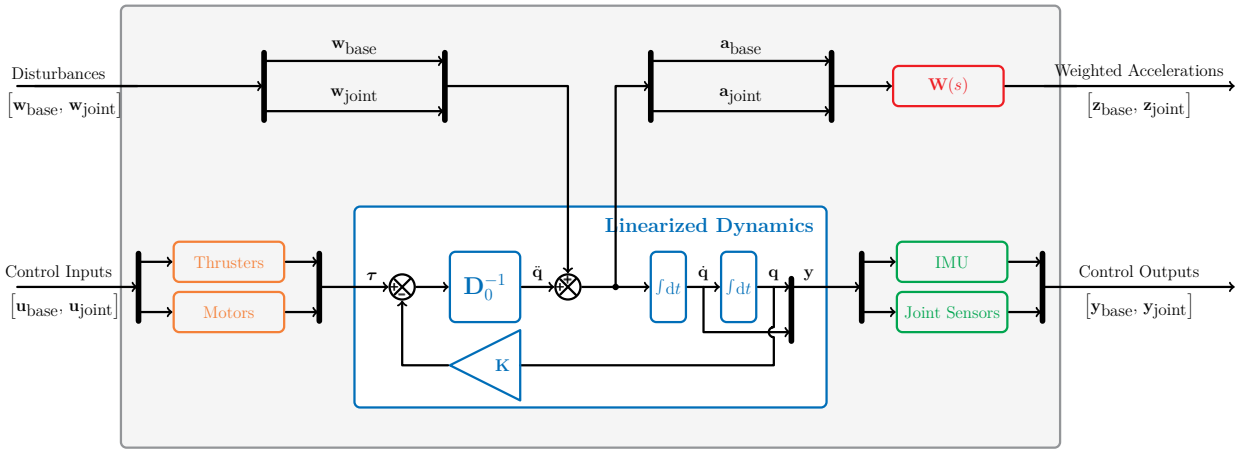


Figure 8.4: H_∞ synthesis scheme

acceleration sensitivity weighting is illustrated in the global synthesis scheme in Figure 8.4. Regarding the numerical optimization, this weighting is considered as a "hard" requirement by `systune`, and will be optimized in priority, compared to the secondary goals, called "soft" requirements (Apkarian, 2013). This weighting is defined by the following transfer function:

$$\mathbf{W}(s) = \frac{s^2 + 2\,\xi_{des}\,\omega_{des}\,s + \omega_{des}^2}{s^2} \tag{8.11}$$

characterized by a damping ratio ξ_{des}, and a pulsation ω_{des}, defining the desired dynamics on the acceleration transfer.

Since the H_∞ synthesis aims at minimizing the weighted transfers to fulfill the related requirements, this weighting aims at shaping the system acceleration as $1/\mathbf{W}(s)$, or equivalently, to shape the position response as a traditional well-damped second-order system. The same requirements are defined for both controllers, but with a different bandwidth and damping. The spacecraft weighting $\mathbf{W}_\mathrm{b}(s)$ is defined by $(\boldsymbol{\omega}_\mathrm{b}, \xi_\mathrm{b}) = (\boldsymbol{\omega}_\mathrm{t}, 0.707)$, in order to obtain a rather slow closed-loop system. The objective is to minimize the base consumption with limited performances, while still being at least capable of tracking the global rotation of the debris. On the other hand, the arm weighting $\mathbf{W}_\mathrm{m}(s)$ is defined by $(\boldsymbol{\omega}_\mathrm{m}, \xi_\mathrm{m}) = (10\,\boldsymbol{\omega}_\mathrm{t}, 0.707)$, in order to reduce as much as possible the tracking error when following the target. In the sequel, the target angular rate is fixed to $\boldsymbol{\omega}_\mathrm{t} = 4°/s$, as proposed by (Lampariello, 2010).

In addition, a second requirement is added in the synthesis to limit the gain value as much as possible, provided that the "hard" requirement of $\mathbf{W}(s)$ is satisfied. This "soft" requirement is only given by a maximum gain that the controller should not exceed. In the following syntheses, its value is initialized with the diagonal terms of the mass matrix multiplied by

the corresponding desired pulsation and damping. For example, the proportional gains of the decentralized design are initialized with $\omega_{des}^2\,\mathbf{D}_{i,i}$, while the derivative ones are initialized with $2\xi_{des}\omega_{des}\,\mathbf{D}_{i,i}$. For the centralized case, only the terms (ω_{des}^2, $2\xi_{des}\omega_{des}$) are kept. These values are also used during the synthesis to limit the controller gains by the order of magnitude of the system inertias.

Mono/Multi-models synthesis The synthesis aims at providing the best controller to track the capture trajectory. Hence, the synthesis models are computed at given configurations along a pre-computed trajectory. Two options are then possible to tune the controllers: either by using each model separately to obtain the best controller at this exact configuration, or by performing the H_∞ optimization with a multi-model objective satisfied by a single controller. This latter approach provides a unique controller for the whole trajectory and thus reduces dramatically the on-line computational burden by avoiding the controller scheduling or the use of nonlinear methods. Using the `systune` function, the weighting functions are specified as "hard constraints", while a gain limitation and a consumption minimization are set by "soft constraints" (Apkarian, 2013).

8.2.4 Results of synthesis

The synthesis is performed for the space robot model presented in Appendix D, with the arm designed by MDA in D.1.3 and the base data in D.2.1. To improve the robust performance of the controller, several models obtained along a capture trajectory are simultaneously considered in the synthesis; a common controller is then synthesized by the multi-model optimization. These results are first presented in the rigid case, and then, in the flexible case. Nevertheless, this latter happens to be unstable with the controller designed on the rigid models, such that a new synthesis is necessary. For the sake of brevity, the gains are presented thoroughly only for the decentralized strategy since fewer terms are computed. The centralized case is covered more quickly, to rather focus on the problems encountered for the flexible system.

8.2.4.1 Rigid space robot

Two different configurations were used to compare the mono and multi-model syntheses. The syntheses are first performed with the decentralized controller of the arm. A space robot configuration $\mathbf{q}$ is described, on the one hand, by the spacecraft states $\mathbf{q}_\mathrm{b}$, and on the other hand, by the arm states $\mathbf{q}_\mathrm{m}$. The space inertial position and attitude in $\mathbf{q}_\mathrm{b}$ have no influence

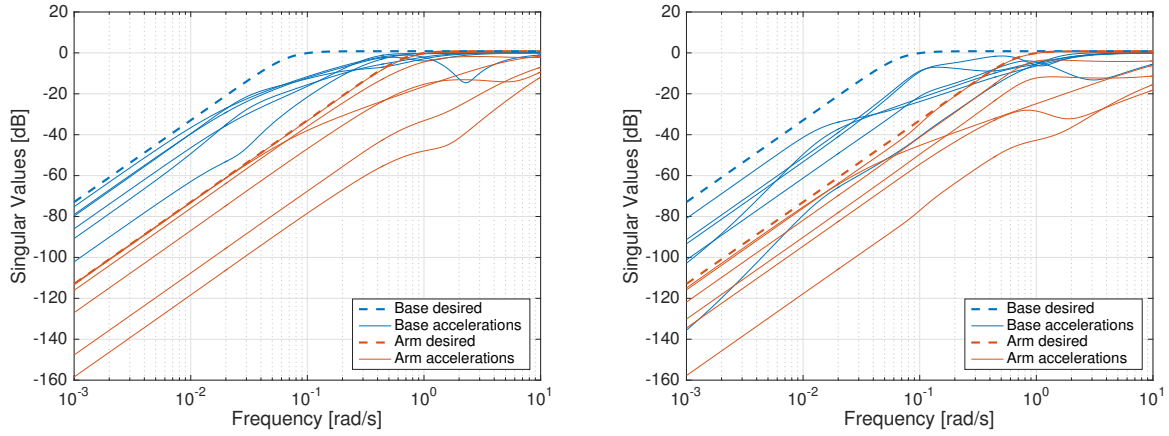


Figure 8.5: Frequency responses on the acceleration sensitivity function for the rigid decentralized arm controller; (a) Mono-model approach; (b) Multi-model approach.

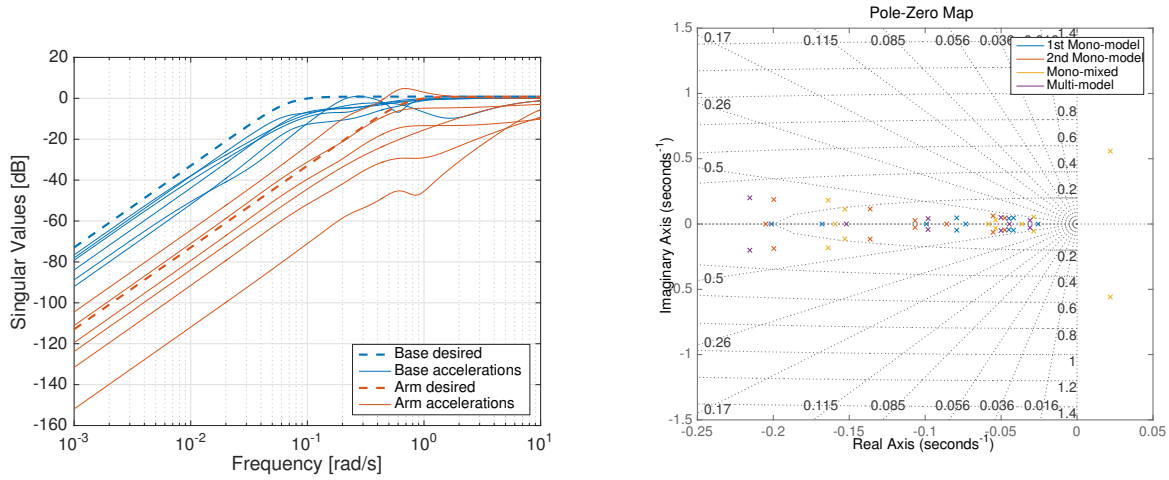
on the global mass matrix in (6.48), since this latter is expressed in the body-fixed frame $\mathcal{R}_{\mathrm{b}}$. Therefore, the model used for synthesis only depends on the arm states. In the following discussion on mono and multi-model syntheses, two random configurations are taken, far from one another, in order to better assess the validity domain of the local controller.

The numerical values used for both configurations are given by:

$$\begin{cases} \mathbf{q}_{\mathrm{m}_1} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top \\ \mathbf{q}_{\mathrm{m}_2} &= \begin{bmatrix} 2.21 & 0.81 & 1.27 & 1.64 & 2.83 & 1.25 \end{bmatrix}^\top \end{cases} \tag{8.12}$$

Decentralized strategy Using the decentralized arm controller in (8.3), and the AOCS one in (8.1), the mono-model synthesis provides the gains summarized in Table 8.2, and the frequency responses obtained for the acceleration sensitivity function are illustrated in Figure 8.5a for the first case. For the different configurations, the requirement is fulfilled with an H_∞ index γ_{hard} and a soft index γ_{soft} both smaller than 1, as shown in Table 8.2. It is also seen on the frequency response that the transfers are below the reference weighting functions of the base and of the arm.

Nevertheless, when the same controller is used for different configurations, the requirements are no longer met and the system even becomes unstable, as illustrated in Figures 8.6a and 8.6b. It means that the H_∞ controller obtained for a given configuration is no longer acceptable for some other configurations. This result is not obvious for close arm config-



(a) (b)

Figure 8.6: Analysis of the rigid decentralized arm controller used on a different arm configuration; (a) Frequency responses on the acceleration sensitivity function; (b) Pole-zero map.

urations, but rather appears for substantially different ones. A result of Lyapunov theory will later show that a positive-definite gain matrix on the whole proportional and derivative actions on both the base and the manipulator would avoid this destabilization. It only appears when the H_∞ synthesis provides a non positive-definite gain matrix. The mono-model controller needs thus to be updated to remain valid on the whole configuration envelope of the capture trajectory.

This is done using the multi-model approach, where the different synthesis models are merged together at the optimization step in `systune`. The algorithm tries to find a unique optimal controller meeting the requirements on all the models simultaneously. For the present capture trajectory, this approach provides a final gamma $\gamma_{hard} < 1$, and yields the frequency responses

Table 8.2: Gains setting of the decentralized arm controller

Model	Gains	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	γ_{hard}	γ_{soft}
Mono 1	$\boldsymbol{K}_p$	2821	3735	3876	5872	1242	0.002	0.97	0.426
	$\boldsymbol{K}_d$	2215	571	594	445	338	0.03		
Mono 2	$\boldsymbol{K}_p$	1025	2103	1634	266	2018	0.008	0.97	0.52
	$\boldsymbol{K}_d$	371	1335	1090	415	289	0.03		
Multi	$\boldsymbol{K}_p$	3803	7246	5575	373	3967	0.006	0.99	0.567
	$\boldsymbol{K}_d$	1973	2507	1822	1222	898	0.03		

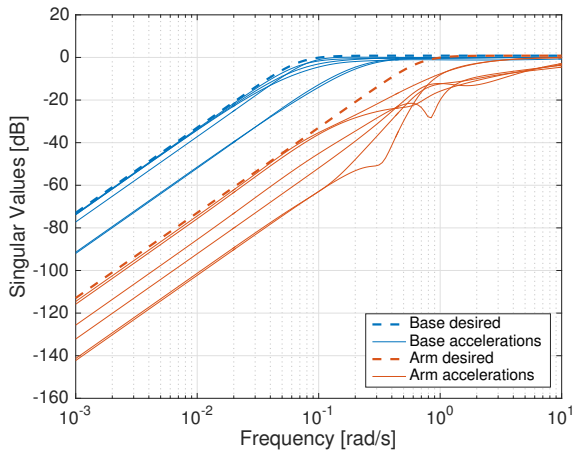


Figure 8.7: Frequency responses on the acceleration sensitivity function for the rigid centralized arm controller with the mono-model approach.

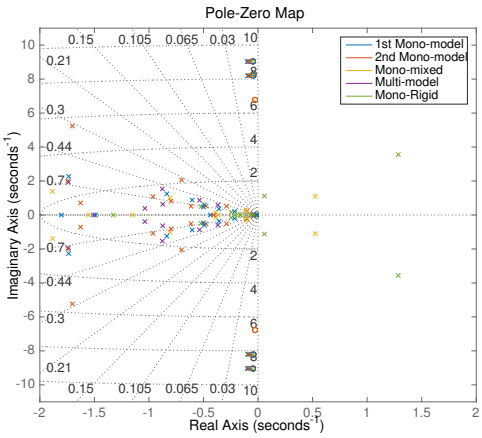


Figure 8.8: Pole-zero map of the rigid decentralized arm controller used on the corresponding flexible space robot model.

presented in Figure 8.5b. It clearly appears that the requirements are fulfilled with the new controller proposed in Table 8.2, but at the expense of higher gains. Indeed, the soft index γ_{soft} is greater for the multi-model synthesis, meaning that the optimization led to higher gains in order to maintain the controller performances with different plants.

Centralized strategy The same kind of conclusions holds for the centralized arm controller. The main advantage of using this controller is a better decoupling of the arm dynamics by the use of the mass matrix inside the controller structure. As illustrated in Figure 8.7, the frequency responses on the acceleration sensitivity function fulfill perfectly the weightings. Nevertheless, this is at the expense of a "specialization" of the controller for the current configuration, and its domain of validity shrinks even more. Concerning the multi-model, the optimization still manages to find a common controller valid for both models.

The discussion about these two types of controllers is rather led in the numerical simulation section, where their pros/cons are emphasized by the accuracy of the end-effector positioning and by the fuel consumption of the base.

8.2.4.2 Flexible space robot

The addition of flexible elements presented in the modeling section can lead to instability with the previous controllers. This result rises the fundamental question of whether to take the flexible modeling into account, or not, during the synthesis process. Indeed, when the

previous controller based on the rigid model of the arm is applied on the flexible one, then the system can become unstable, as shown in Figure 8.8.

The same arm configuration is taken from the previous section in (8.12). It is important to see that these values define the rigid state of the arm, while the extra flexible states define the deformed shape of the segments. For the synthesis purpose, all these flexible variables are set to zero, corresponding to the undeformed state of the arm. However, their dynamics is taken into account through the augmented mass matrix with the flexible sub-blocks. From the arm design provided by MDA in Appendix D.1.3, the flexible frequencies obtained by the AMM method introduced in appendix B.1 are presented in Table 8.3. Six modes are used to describe the two main flexible segments of the arm. One is used for the traction, one for the torsion, and two in each bending direction. They are described by their modal shapes denoted by $\boldsymbol{\phi}_{x/y/z/\alpha}$, and by their natural frequencies denoted by $\boldsymbol{\omega}_{x/y/z/\alpha}$ in rad/s. It is important to recall that these modes are the natural frequencies of the isolated segments. They will slightly shift when looking at the global manipulator dynamics, and even more for the space robot ones, because of the coupling between the segments and with the other appendages. A first glance at these values shows that traction is, as expected, described by a very high frequency mode and could be neglected in the simulations. In addition, the bending modes yield different frequencies because the load is asymmetric for this arm, as opposed to the Canadarm design presented in Section 5.2.6.1.

The flexible modes frequencies for the solar arrays are also provided in appendix D.2.1, and recalled in Table 8.3. They are set to slightly different values in order to identify them in the frequency responses.

Table 8.3: Natural flexible frequencies of the robotic arm and of the solar arrays (in rad/s)

Joint	$\boldsymbol{\omega}_x$	$\boldsymbol{\omega}_{y,1}$	$\boldsymbol{\omega}_{y,2}$	$\boldsymbol{\omega}_{z,1}$	$\boldsymbol{\omega}_{z,2}$	$\boldsymbol{\omega}_\alpha$
2	181.74	2.26	10.01	2.29	10.05	15.21
3	189.27	4.60	45.72	4.40	24.35	14.83
Solar Array		$\boldsymbol{\omega}_y$		$\boldsymbol{\omega}_z$		$\boldsymbol{\omega}_\alpha$
Left		8		15		20
Right		8.8		16.5		22

Decentralized strategy Using the same decentralized controller for the arm in (8.3) and the AOCS controller in (8.1), the syntheses are performed again for the mono and multi-model approaches. The resulting gains and optimization indices are given in Table 8.4. The main result is that the gains become much more important; this is shown with the soft index

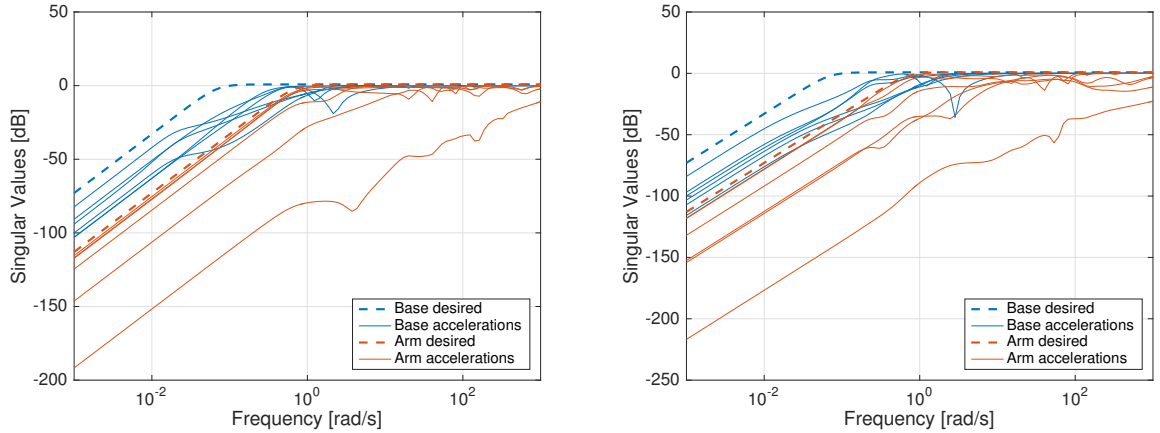


Figure 8.9: Frequency responses on the acceleration sensitivity function for the flexible decentralized arm controller; (a) Mono-model approach; (b) Multi-model approach.

γ_{soft} reaching the value of 100. One solution may be to add some dynamics to the arm controller by means of first order terms or even more advanced filters, like notch filters, in order to manage the flexible modes and to add roll-off at high frequencies.

As mentioned above, the multi-model synthesis leads to a valid and efficient controller on a greater configuration envelope. Again, the use of a controller tuned on one model leads to instability when applied on a different arm configuration.

Centralized strategy The flexible syntheses lead to the same conclusion for the centralized controller: one controller tends to destabilize the plant for a closer arm configuration. Its robustness w.r.t. the arm configuration is poorer than the decentralized controller. Nevertheless, since the controller is pre-multiplied by the mass matrix, the gains inside are less important, such that the soft index γ_{soft} is smaller than the decentralized case. It typically reaches values around 10, instead of 100 for the decentralized controller.

Table 8.4: Gains setting of the decentralized arm controller for the mono-model approach

Model	Gains	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	γ_{hard}	γ_{soft}
Mono 1	$\boldsymbol{K}_p$	9197	4827	3560	1360	3138	1	0.99	103
	$\boldsymbol{K}_d$	720	720	720	720	720	165		
Mono 2	$\boldsymbol{K}_p$	4111	4384	3762	989	26060	48496	0.99	114
	$\boldsymbol{K}_d$	794	794	794	794	794	19		
Multi	$\boldsymbol{K}_p$	7419	3556	3420	1440	3648	1	0.99	149
	$\boldsymbol{K}_d$	1040	1040	1040	1020	1040	196		

Until now, the controllers have been designed on a reference plant linearized around a given arm configuration. The robustness of the resulting controller was improved by including more models in the synthesis, but this approach does not guarantee the robust stability and performance of the controller on the whole capture trajectory, or even on a continuous subset of the arm configuration. The next section investigates the LFT modeling of a manipulator w.r.t. the joint angles. It allows to perform a stability and performance μ-analysis on the closed-loop system.

8.3 Robust Analysis

The linear dynamic model used in the synthesis is evaluated at each different configuration of the arm, in order to fulfill the frequency requirement all along the expected trajectory. Another way of studying the robustness of a controller is to model its dependency upon the variable parameters under a LPV form, as the LFT. An example is provided in (Manceaux-Cumer and Chrétien, 2001; Guy et al., 2014) to parametrize the model of a satellite w.r.t. the tilt angle of its solar panel. Doing this, the robust performance of the closed-loop system can be studied and validated on the whole range of variation of the parameter based on the μ-analysis. The main requirement for this modeling is that the uncertainties dependency can be set in a linear form to extract them from the nominal plant.

The common LFT model of a system is illustrated in Figure 8.10a, where the nominal plant is denoted by $\mathbf{\Sigma}$, the controller by $\boldsymbol{C}$, and the uncertainty block by $\mathbf{\Delta}$. The system references and the regulated outputs are denoted by ($\mathbf{w}$, $\mathbf{z}$) and represent the exogenous signals. Then, the controller loop is based on the measurements $\mathbf{y}$ and provides the commands $\mathbf{u}$, while the

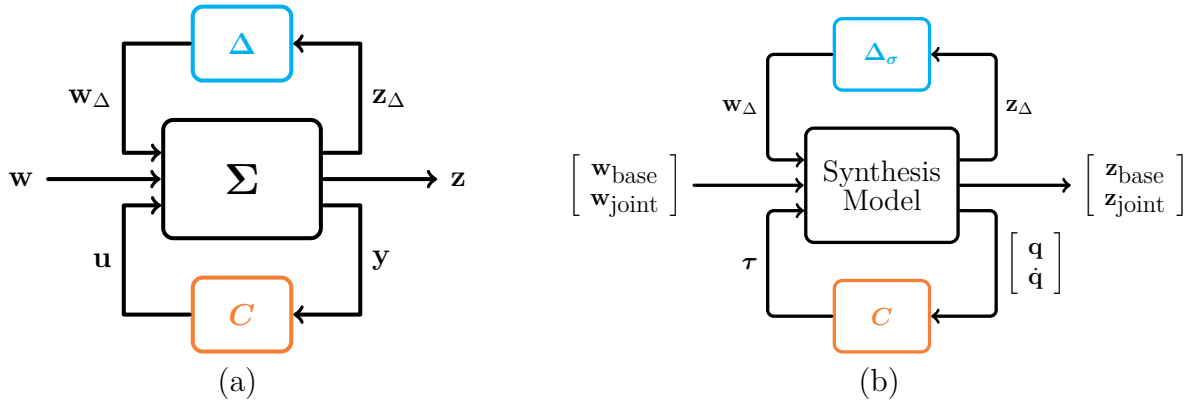


Figure 8.10: LFT model of an uncertain system; (a) General LFT form; (b) LFT form of the linearized model of the robotic system.

outputs multiplying the uncertainty block are denoted by $\mathbf{z}_\Delta$ and the resulting inputs by $\mathbf{w}_\Delta$. For the case of the mass matrix of a robotic system, either a manipulator or a space robot, the scheme is written explicitly w.r.t. the previous notations and to the inner loop in Figure 8.1. The dependency of the mass matrix upon the joint angles in $\mathbf{q}$ must be parametrized in a rational way to obtain its LFT form. To do this, an intermediate variable, denoted $\boldsymbol{\sigma}$, is defined to express this complex dependency in a linear way.

The dependency of the mass matrix upon the joint angles $\mathbf{q}$ is first written to emphasize that the rotation matrices are the only terms that need to be parameterized. Then, the LFT model of a single rotation matrix is proposed in two ways, before applying it to the global mass matrix. The μ tools provided by the French Aerospace Laboratory (ONERA) are finally used to model the whole space robot and to perform the robust analysis of the previous controllers (Roos, 2013).

8.3.1 Robotic arm parameterization

The results presented in Appendix C.1.1 are recalled here to derive explicitly the dependency of $\mathbf{D}$ in $\mathbf{q}$. Using the closed-form equation, the general form of the mass matrix is taken from (6.52). It reads as follows for any of the mechanical system considered in the thesis, either a manipulator or a space robot, with rigid or flexible segments:

$$\begin{aligned}\mathbf{D} &= \mathbf{N}^\top\ [\mathbf{M}]\ \mathbf{N} \\ &= \mathbf{N}_d^\top\ \mathbf{N}_l^\top\ [\mathbf{M}]\ \mathbf{N}_l\, \mathbf{N}_d\end{aligned}$$

In this expression, the segment mass matrices in $[\mathbf{M}]$ and the DeNOC matrix $\mathbf{N}_d$ are given by :

$$[\mathbf{M}] = \text{diag}(\mathbf{M}_i\ ,\ i = 0 \dots n)$$
$$\mathbf{N}_d = \text{diag}(\mathbf{P}_i\ ,\ i = 0 \dots n)$$

where the index 0 stands for the base variables.

Since these matrices are expressed in the local segment frame only, they are defined either by dynamic or kinematic parameters. Hence, they are constant and independent of the system

configuration. The $\mathbf{q}$-dependency is entirely contained in the matrix $\mathbf{N}_l$, which reads:

$$\mathbf{N}_l = \left[\begin{array}{ccccc} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{2,1} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{3,1} & \mathbf{A}_{3,2} & \mathbf{I} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{A}_{n,1} & \mathbf{A}_{n,2} & \cdots & \mathbf{A}_{n,n-1} & \mathbf{I} \\ \hline \mathbf{A}_{n+1,1} & \mathbf{A}_{n+1,2} & \cdots & \mathbf{A}_{n+1,n-1} & \mathbf{A}_{n+1,n} \end{array}\right]$$

where the notation of the robotic manipulators is used, recalling that it has been extended to the space robot by only specifying the appendage number on the indices (i, j).

In this expression, the twist-propagation matrices were introduced for the rigid and flexible cases by decoupling the rotation matrix from the kinematic part in (4.30a) and (5.45a), as follows:

$$\mathbf{A}_{i,i-1} = {}^{i-1}\boldsymbol{R}_i^{\top\ (i-1)}\mathbf{A}^{\mathrm{r/f}}_{i,i-1} \tag{8.13}$$

where the rigid version of the twist-propagation matrix $\mathbf{A}^{\mathrm{r}}_{i,i-1}$ is constant and only function of the segment kinematics, while the flexible version $\mathbf{A}^{\mathrm{f}}_{i,i-1}$ was split into a nominal term, which is constant, and first order term w.r.t. the flexible variables.

The general form of the augmented rotation matrix ${}^{i-1}\boldsymbol{R}_i$ is introduced in (4.31) and (5.46) using the same formalism for the rigid and flexible cases, as follows:

$${}^{i-1}\boldsymbol{R}_i = \left[\begin{array}{ccc} {}^{i-1}R_i^{\mathrm{r/f}} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times n_{f_i}} \\ \mathbf{0}_{3\times 3} & {}^{i-1}R_i^{\mathrm{r/f}} & \mathbf{0}_{3\times n_{f_i}} \\ \mathbf{0}_{n_{f_i}\times 3} & \mathbf{0}_{n_{f_i}\times 3} & \mathbf{0}_{n_{f_i}\times n_{f_i}} \end{array}\right]$$

where the flexible rotation matrix can be decoupled as a function of the rigid rotation matrix using (5.33). Its expression was given by the product of the rotation induced by the flexible deflection of the previous segment, with the rotation induced by the rigid body motion at the joint. It is written as:

$${}^{i-1}R_i^{\mathrm{f}} = {}^{\mathrm{o}_{i-1}}R_{\mathrm{e}_{i-1}}(\boldsymbol{\delta}_{i-1})\ {}^{i-1}R_i^{\mathrm{r}}(\theta_i)$$

where the flexible rotation ${}^{\mathrm{o}_{i-1}}R_{\mathrm{e}_{i-1}}(\boldsymbol{\delta}_{i-1})$ reduces to $\mathbf{0}$ since the coordinates $\boldsymbol{\delta}_i$ are assumed to be zero when linearizing the dynamic model in (8.7).

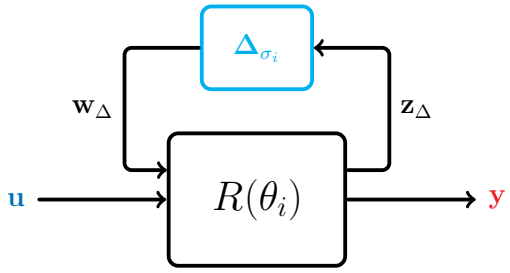


Figure 8.11: LFT model of a rotation w.r.t. a varying angle θ_i parameterized by σ_i

As a conclusion, the only terms of the mass matrix that explicitly depend in the coordinates $\mathbf{q}$ are the rigid rotation matrices ${}^{i-1}R_i^{\mathrm{r}}$. Therefore, the parameterization of a single rotation matrix in the joint angle θ allows to parameterize the whole linearized model in $\mathbf{q}$. In the next section, two parameterizations are presented to achieve this goal.

8.3.2 LFT parameterization of a rotation matrix

The rotation matrices are always expressed along one principal axis (x, y, z), and any other complex rotation can be derived from these elementary versions. Denoting by (c_θ, s_θ) the cosine/sinus of the angle θ, they read as follows (Craig, 1989):

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\theta & -s_\theta \\ 0 & s_\theta & c_\theta \end{bmatrix} \qquad R_y(\theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \qquad R_z(\theta) = \begin{bmatrix} c_\theta & -s_\theta & 0 \\ s_\theta & c_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Following the control theory convention of notation, any of these matrices can be viewed as a system whose the input is the expression of a vector in the frame $\mathcal{R}_{in}$, denoted $\mathbf{u}$, and the output is its expression in the frame $\mathcal{R}_{out}$, as illustrated in Figure 8.11. The change of frame for the rotation along the Z-axis is then given by:

$$\mathbf{y} = {}^{out}R_{in}\,\mathbf{u} \quad \Longrightarrow \quad \begin{cases} y_1 &= \cos(\theta)\,u_1 - \sin(\theta)\,u_2 \\ y_2 &= \sin(\theta)\,u_1 + \cos(\theta)\,u_2 \\ y_3 &= u_3 \end{cases} \tag{8.14}$$

For the two other rotations, the same equations holds, provided that one re-orders the indices accordingly.

The LFT model of the rotation matrix is then obtained by finding a parameter σ that can express the previous equation using a linear scheme, i.e., to express the blocks $\mathbf{\Sigma}$ and $\mathbf{\Delta}$ of

the general form. In the first case, the tangent of the half-angle is used based on the works of (Manceaux-Cumer and Chrétien, 2001), and in the second one, the quarter of the angle is used to rewrite these expressions in a minimal way (Guy et al., 2014). In the following, the quantities are defined with the subscript "2" and "4" to distinguish between the half-angle parameterization or the quarter-angle one. It is important to keep in mind that the LFT models are numerically tractable for a limited number of variables, so the fewer the repetitions of the uncertain parameters, the better the results obtained by the μ-analysis.

8.3.2.1 Parameterization with $\tan(\theta/2)$

As done in (Manceaux-Cumer and Chrétien, 2001), the previous set of equations involved in the rotation matrix are re-written as a function of the tangent of the half-angle. The corresponding uncertain parameter reads:

$$\sigma_2 \triangleq \tan\left(\theta/2\right) \tag{8.15}$$

Using the classic trigonometric relations, the cosine and the sinus can be expressed as functions of the tangent of the half-angle, as follows:

$$\cos(\theta) = \frac{1-\tan^2(\theta/2)}{1+\tan^2(\theta/2)} = \frac{1-\sigma_2^2}{1+\sigma_2^2} \tag{8.16}$$

$$\sin(\theta) = \frac{2\tan(\theta/2)}{1+\tan^2(\theta/2)} = \frac{2\sigma_2}{1+\sigma_2^2} \tag{8.17}$$

The advantage of this formulation is to shift the uncertainty from the angle θ, and its cosine/sinus, to the tangent of its half by means of rational functions. This transformation allows to model the uncertainty using the LFT framework.

The change of basis given in (8.14) can be re-written as follows:

$$\boldsymbol{y} = {}^{out}R_{in}\,\boldsymbol{u} \quad \Longrightarrow \quad \left\{ \begin{array}{rcl} y_1 & = & \dfrac{1-\sigma_2^2}{1+\sigma_2^2}\,u_1 - \dfrac{2\sigma_2}{1+\sigma_2^2}\,u_2 \\ y_2 & = & \dfrac{2\sigma_2}{1+\sigma_2^2}\,u_1 + \dfrac{1-\sigma_2^2}{1+\sigma_2^2}\,u_2 \\ y_3 & = & u_3 \end{array} \right.$$

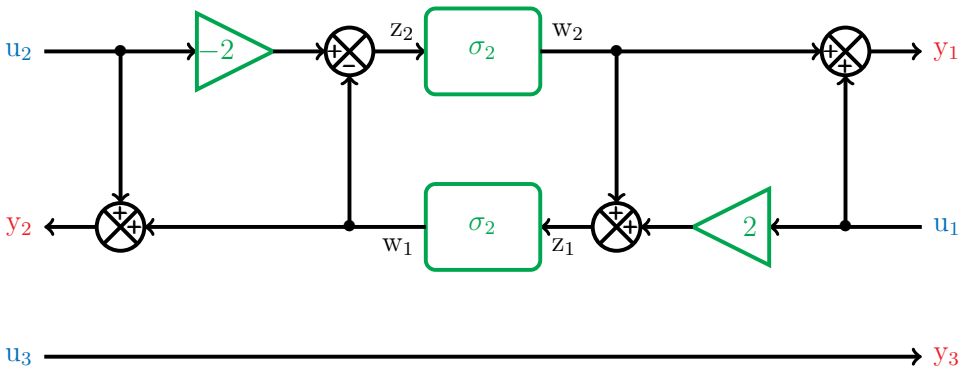


Figure 8.12: LFT scheme to parameterize the rotation θ along Z with $\sigma_2 = \tan(\theta/2)$

where each equation is re-arranged to make common terms appear:

$$y_1 = u_1 - \frac{2\sigma_2^2}{1+\sigma_2^2}\,u_1 - \frac{2\sigma_2}{1+\sigma_2^2}\,u_2 = u_1 - 2\,\frac{\sigma_2}{1+\sigma_2^2}\,(\sigma_2\,u_1 - u_2)$$

$$y_2 = u_2 + \frac{2\sigma_2}{1+\sigma_2^2}\,u_1 - \frac{2\sigma_2^2}{1+\sigma_2^2}\,u_2 = u_2 + 2\,\frac{\sigma_2}{1+\sigma_2^2}\,(u_1 - \sigma_2\,u_2)$$

$$y_3 = u_3$$

With this formulation, the term $\sigma_2/(1+\sigma_2^2)$ appears in both equations and a feedback loop can easily represent it. Doing this, the LFT representation of the whole rotation matrix can be realized with only two repetitions of the parameter σ_2, by cleverly defining the input/output signals at the appropriate places. A linear scheme is derived in Figure 8.12 to express the previous equation with a minimum number of repetitions. The corresponding uncertainty block introduced in Figure 8.11 is then a 2×2 block $\mathbf{\Delta}_{\sigma_2} = \sigma_2 \mathbf{I}_2$, which is lower that the order 4 obtained in (Manceaux-Cumer and Chrétien, 2001) and in (Guy et al., 2014).

In Figure 8.12, the endogenous inputs and outputs of the general LFT form are introduced by:

$$\mathbf{w}_\Delta = \begin{bmatrix} w_1 & w_2 \end{bmatrix}^\top \qquad\qquad \mathbf{z}_\Delta = \begin{bmatrix} z_1 & z_2 \end{bmatrix}^\top \tag{8.18}$$

They allow to derive the blocks defining the linear model of the rotation matrix under the

LFT form:

$$\mathbf{\Sigma}_{\sigma_2}\left\{\begin{bmatrix}\mathrm{z}_1\\ \mathrm{z}_2\\ \hline \mathrm{y}_1\\ \mathrm{y}_2\\ \mathrm{y}_3\end{bmatrix} = \left[\begin{array}{cc|ccc} 0 & 1 & 2 & 0 & 0\\ -1 & 0 & 0 & -2 & 0\\ \hline 0 & 1 & 1 & 0 & 0\\ 1 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{array}\right] \begin{bmatrix}\mathrm{w}_1\\ \mathrm{w}_2\\ \hline \mathrm{u}_1\\ \mathrm{u}_2\\ \mathrm{u}_3\end{bmatrix} \triangleq \left[\begin{array}{c|c}\Sigma_{11} & \Sigma_{12}\\ \hline \Sigma_{21} & \Sigma_{22}\end{array}\right] \left[\begin{array}{c}\mathbf{w}_\Delta\\ \hline \mathbf{u}\end{array}\right] \tag{8.19}$$

$$\mathbf{\Delta}_{\sigma_2}\left\{\begin{bmatrix}\mathrm{w}_1\\ \mathrm{w}_2\end{bmatrix} = (\sigma_2\,\mathbf{I}_2)\begin{bmatrix}z_1\\ z_2\end{bmatrix} \triangleq \Delta_{\sigma_2}\,\mathbf{z}_\Delta \tag{8.20}$$

The dependency of order 2 is represented by the size of the uncertain block $\mathbf{\Delta}$. For the whole mass matrix, this block will merge the uncertainty related to each joint angle and its size must be as small as possible.

However, for the general case of $\theta \in \,]-\pi;\,\pi]$, the parameter σ_2 varies in $]-\infty;+\infty[$. It may prevent the stability analysis based on the μ-analysis to be performed for these extreme variations. In order to reduce the variation range of $\sigma_2 \in [0;\,+\infty[$, an intermediate variable $\delta_2 \in [0;\,1[$, defined by the bijective relation $\sigma_2 = \delta_2/(1-\delta_2)$, is introduced in (Manceaux-Cumer and Chrétien, 2001). But this relation is only valid for $\sigma_2 \in [0;\,+\infty[$ and does not cover a whole revolution of the joint angle θ.

A workaround is available by simply limiting angle range to avoid these extreme values. A second option is to use the next parameterization of order 4, more complex but valid on the whole range $]-\pi;\,\pi]$. There is always a compromise between the size of the block $\mathbf{\Delta}$ and the accuracy of the μ-analysis.

8.3.2.2 Parameterization $\tan(\theta/4)$

N. Guy introduced another parameterization in (Guy et al., 2014) to overcome this limitation by using the tangent of the quarter-angle. The following trigonometric relation is used to further develop the tangent of the half-angle in the previous model:

$$\sigma_2 = \tan(\theta/2) = \frac{2\tan(\theta/4)}{1-\tan^2(\theta/4)} \triangleq \frac{2\sigma_4}{1-\sigma_4^2}$$

The cosine/sinus functions are then re-parameterized as follows:

$$\cos(\theta) = \frac{1-\sigma_2^2}{1+\sigma_2^2} = \frac{\left(1+\sigma_4^2\right)^2 - 8\sigma_4^2}{\left(1+\sigma_4^2\right)^2} \qquad \sin(\theta) = \frac{2\sigma_2}{1+\sigma_2^2} = \frac{4\sigma_4\left(1-\sigma_4^2\right)}{\left(1+\sigma_4^2\right)^2}$$

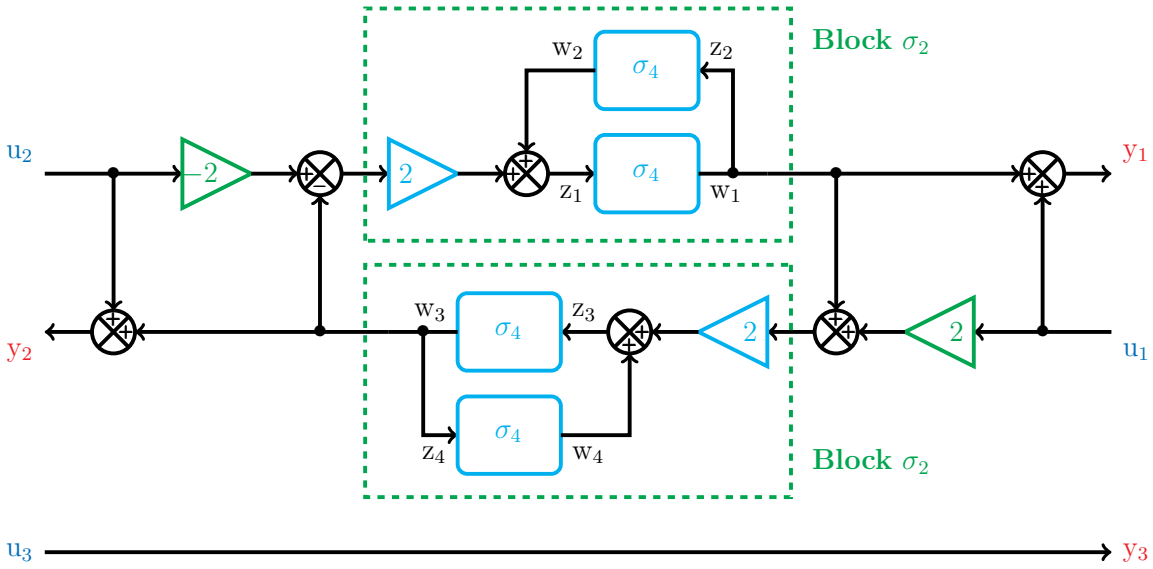


Figure 8.13: LFT scheme to parameterize the rotation θ along Z with $\sigma_4 = \tan(\theta/4)$

The best way to represent the rotation matrix scheme with the parameter σ_4 is to replace the blocks σ_2 in Figure 8.12. Since this model was already minimal, the minimal representation of σ_2 as a function of σ_4 preserves this minimality. Thanks to the previous trigonometric relation, a feedback loop with σ_4 is obtained to represent the block σ_2. The final scheme is illustrated in Figure 8.13.

The endogenous inputs and outputs are augmented by:

$$\mathbf{w}_\Delta = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 \end{bmatrix}^\top \qquad \mathbf{z}_\Delta = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix}^\top \tag{8.21}$$

to finally define the matrices of the LFT model in Figure 8.13, as follows:

$$\mathbf{\Sigma}_{\sigma_4} \left\{ \left[\begin{array}{c} z_1 \\ z_2 \\ z_3 \\ z_4 \\ \hline y_1 \\ y_2 \\ y_3 \end{array}\right] = \left[\begin{array}{cccc|ccc} 0 & 1 & -2 & 0 & 0 & -4 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \left[\begin{array}{c} w_1 \\ w_2 \\ w_3 \\ w_4 \\ \hline u_1 \\ u_2 \\ u_3 \end{array}\right] \triangleq \left[\begin{array}{c|c} \Sigma_{11} & \Sigma_{12} \\ \hline \Sigma_{21} & \Sigma_{22} \end{array}\right] \left[\begin{array}{c} \mathbf{w}_\Delta \\ \hline \mathbf{u} \end{array}\right] \tag{8.22}$$

$$\mathbf{\Delta}_{\sigma_4} \left\{ \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = (\sigma_4 \, \mathbf{I}_4) \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \triangleq \Delta_{\sigma_4} \, \mathbf{z}_\Delta \right. \tag{8.23}$$

Thanks to the first model based on the parameter σ_2, the whole rotation matrix is represented with an uncertain block of order 4 only, which is better than the order 8 introduced in (Guy et al., 2014). The main idea in this derivation is to minimize the repetition of the original block σ_2 in the scheme of Figure 8.12, in order to minimize the repetition of the new parameter σ_4. Finally, the LFT model provided in (8.22) and (8.23) performs the rotation of the vector $\mathbf{u}$ as the rotation matrix ${}^{out}R_{in}$ does. Even though the uncertainty block Δ_{σ_4} is now of size 4, it stays valid on the whole range of variation $\theta \in]-\pi;\, \pi]$.

8.3.3 Robust stability analysis

The previous LFT model of a rotation matrix is used in the algorithm that computes the mass matrix presented in Appendix C.1. The basic operations like the summation, the inversion or the feedback of the linear system must be extended to account for LFT models instead of nominal models. They are implemented in the efficient toolbox *SMAC* developed by the French Aerospace Laboratory, also called *ONERA*. This tool is available on the institution website[2], and provides many functions for the modeling, the analysis and the control of uncertain systems. In the sequel, only the modeling and analysis tools are used with the *GSS* library to represent the uncertain systems, and the *SMART* toolbox to analyze their robust stability and performance. Each of these libraries possess its own functions dedicated to the μ-analysis.

First, this section assesses the robust stability of a given controller by confronting the numerical results obtained by μ-analysis to the Lyapunov theory. Indeed, it can be shown that a PD controller guarantees the asymptotic stability of the linearized system, provided that the gains matrices are positive-definite. On the contrary, the numerical tools proved to perform poorly when applied on the whole range of variation of the joint angles. They bring a guarantee of stability on a given subset of the range variation though, and can be refined by performing the analyses on a reduced frequency range.

Then, the robust performances are studied in the second section. The SMART tools are applied to the augmented model with the performance transfer on the acceleration, as illustrated in Figure 8.4. As it is the case for the stability analysis, the guarantee can only be reached for subsets of the whole range of joint configurations. Though, the derived boundaries remains useful to either assess the robust performance of a given controller, or to extract a worst-case configuration. This latter can then be included in an iterative multi-model synthesis in order to provide a more robust controller.

[2] http://w3.onera.fr/smac/

8.3.3.1 Lyapunov analysis of the stability

Based on the Lyapunov theory (Khalil, 2001), the stability of a system described by the states $\boldsymbol{x}$ and by the following dynamic equation:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) \tag{8.24}$$

can be assessed around an equilibrium point at the origin using a *Lyapunov candidate* $V(\boldsymbol{x})$. This latter is defined as a continuously differentiable function of scalar values on a subset D containing the origin ($\mathbf{0} \in D$), denoted $V : D \subset \mathbb{R}^n \longmapsto \mathbb{R}$, and satisfying the the positive definiteness w.r.t. $\boldsymbol{x}$:

$$V(\mathbf{0}) = 0 \qquad \text{and} \qquad V(\boldsymbol{x}) > 0, \quad \forall \boldsymbol{x} \neq \mathbf{0} \tag{8.25}$$

With these assumptions, the following theorem is used in the sequel ((Khalil, 2001), Theorem 4.1, p.114):

$$\boxed{\text{If } \dot{V}(\boldsymbol{x}) \leq 0 \text{ in } D, \text{ then } \boldsymbol{x} = \mathbf{0} \text{ is stable.}} \tag{8.26}$$

In addition to this conclusion, if the time derivative of the Lyapunov candidate is negative-definite, then the origin is *asymptotically stable*.

As demonstrated in (Spong et al., 2006), the basic relation to prove the asymptotic stability by means of the Lyapunov theory is first to obtain the skew-symmetry property of the matrix defined by:

$$S(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{D}}(\mathbf{q}) - 2\,\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \tag{8.27}$$

when coming back to the modeling definition of both matrices, one obtains from (6.48) and (6.49) that:

$$\begin{aligned} S(\mathbf{q}, \dot{\mathbf{q}}) &= \dot{\mathbf{D}}(\mathbf{q}) - 2\,\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \\ &= \left(\dot{\mathbf{N}}^\top\,[\mathbf{M}]\,\mathbf{N} + \mathbf{N}^\top\left[\dot{\mathbf{M}}\right]\mathbf{N} + \mathbf{N}^\top\,[\mathbf{M}]\,\dot{\mathbf{N}}\right) - 2\left(\mathbf{N}^\top\,[\mathbf{M}]\,\dot{\mathbf{N}} + \mathbf{N}^\top\left[\dot{\mathbf{M}}\right]\mathbf{N}\right) \\ &= -\mathbf{N}^\top\left[\dot{\mathbf{M}}\right]\mathbf{N} + \left(\dot{\mathbf{N}}^\top\,[\mathbf{M}]\,\mathbf{N} - \mathbf{N}^\top\,[\mathbf{M}]\,\dot{\mathbf{N}}\right) \\ &= -\mathbf{N}_d^\top\,\dot{\widehat{\mathbf{M}}}\,\mathbf{N}_d - \mathbf{N}_d^\top\left([\boldsymbol{\Omega}]\,\widehat{\mathbf{M}} + \mathbf{H}^\top - \mathbf{H} - \widehat{\mathbf{M}}\,[\boldsymbol{\Omega}]\right)\mathbf{N}_d \end{aligned}$$

The two terms $(\dot{\widehat{\mathbf{M}}}, \mathbf{H})$ were introduced to derive the dynamic algorithm of the Coriolis and centrifugal matrix in appendix A.3.3. The RHS of the equation is obviously skew-symmetric,

but the proof for the first part is more involved and would need to come back to the explicit definition of $\dot{\mathbf{M}}_i$ to derive it. The complete proof of this property is available in the general case in (Spong et al., 2006), and the result is used in the sequel.

As proposed in (Spong et al., 2006), for a PD controller of the form $\boldsymbol{\tau} = -\boldsymbol{K}_p\tilde{\mathbf{q}} - \boldsymbol{K}_d\dot{\mathbf{q}}$, with $\tilde{\mathbf{q}} = (\mathbf{q}_d - \mathbf{q})$ the position error, the following Lyapunov candidate is used to prove the stability with $\boldsymbol{x} = \left[\ \tilde{\mathbf{q}}^\top, \dot{\mathbf{q}}^\top\ \right]^\top$:

$$V(\boldsymbol{x}) = \frac{1}{2}\,\dot{\mathbf{q}}^\top\,\mathbf{D}\,\dot{\mathbf{q}} + \frac{1}{2}\,\tilde{\mathbf{q}}^\top\,\boldsymbol{K}_p\,\tilde{\mathbf{q}} \tag{8.28}$$

The time derivative of this Lyapunov candidate is now derived using the fact that $S(\mathbf{q}, \dot{\mathbf{q}})$ is skew-symmetric and that the control law is given by a PD controller. It yields :

$$\begin{aligned}
\dot{V}(\boldsymbol{x}) &= \dot{\mathbf{q}}^\top\,\mathbf{D}\,\ddot{\mathbf{q}} + \frac{1}{2}\,\dot{\mathbf{q}}^\top\,\dot{\mathbf{D}}\,\dot{\mathbf{q}} + \dot{\mathbf{q}}^\top\,\boldsymbol{K}_p\,\tilde{\mathbf{q}} \\
&= \dot{\mathbf{q}}^\top\,(\boldsymbol{\tau} - \mathbf{C}\dot{\mathbf{q}}) + \frac{1}{2}\,\dot{\mathbf{q}}^\top\,\dot{\mathbf{D}}\,\dot{\mathbf{q}} + \dot{\mathbf{q}}^\top\,\boldsymbol{K}_p\,\tilde{\mathbf{q}} \\
&= \frac{1}{2}\,\dot{\mathbf{q}}^\top\,\left(\dot{\mathbf{D}} - 2\,\mathbf{C}\right)\dot{\mathbf{q}} - \dot{\mathbf{q}}^\top\,\boldsymbol{K}_d\,\dot{\mathbf{q}} \\
&= -\dot{\mathbf{q}}^\top\,\boldsymbol{K}_d\,\dot{\mathbf{q}}
\end{aligned}$$

Two main remarks follow on from this result. First, to ensure that the Lyapunov candidate is valid, the proportional gain matrix must be positive-definite. Secondly, the time derivative of this function is negative-definite if the derivative gain matrix is positive-definite. **Therefore, if a PD controller with positive-definite gain matrices is used on the entire system, then the closed-loop remains asymptotically stable whatever the joint configuration.** This result may seem contradictory with the unstable closed-loop found in the previous section, but it is actually demonstrating that the H_∞ optimization led to PD controllers with non positive-definite gain matrices. It does occur especially when gain matrices are left free in the optimization process, like the 3×3-blocks $\left(\boldsymbol{K}_{p_{pos}}, \boldsymbol{K}_{p_{rot}}\right)$ on the base controller.

8.3.3.2 μ-analysis of the stability

Thanks to the previous uncertain model of a single rotation, the complete parameterization of the mass matrix is derived under a LFT form using the SMAC toolbox of ONERA (Roos, 2013). The model of the rotation is inserted into the algorithm in Algorithm C.1, and the matrices are represented as *GSS* objects, standing for *general state-space*. Each product,

summation or inverse in the algorithm is handled by the SMAC toolbox thanks to the dedicated functions. At the end, a model reduction allows to obtain the minimal uncertain representation of the whole mass matrix, w.r.t. the joint angles.

For the example of the previous space robot, the global model is parameterized by a block $\boldsymbol{\Delta}_\sigma$ of size 40×40 using the parameterization $\sigma_{2_i} = \tan(\theta_i/2)$, and of size 80×80 for the parameterization $\sigma_{4_i} = \tan(\theta_i/4)$. The uncertain block is split as follows for the robotic arm with 6 joints:

$$\boldsymbol{\Delta}_\sigma = \begin{bmatrix} \boldsymbol{\Delta}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Delta}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{\Delta}_5 \end{bmatrix} \tag{8.29}$$

with

$$\boldsymbol{\Delta}_i = \sigma_{2_i}\,\mathbf{I}_8 \qquad \text{or} \qquad \boldsymbol{\Delta}_i = \sigma_{4_i}\,\mathbf{I}_{16} \tag{8.30}$$

It is worth noticing that only the first five joint angles are accounted for in this uncertain block. Indeed, the mass matrix dependency on the sixth joint vanishes because it rotates the last segment along an axis of symmetry, and as a consequence, does not influence the resulting inertia at the base.

In addition, the sizes mentioned above are actually the required number of parameters to represent twice the double rotation matrix, denoted ${}^{i-1}\boldsymbol{R}_i = \text{diag}({}^{i-1}R_i, {}^{i-1}R_i)$. Indeed, each parameter σ_2 is repeated 2 times per rotation matrix so 4 times for this double rotation, respectively 4 times per rotation and 8 times per double rotation for σ_4. Finally, ${}^{i-1}\boldsymbol{R}_i$ multiplies the twist-propagation matrix, which is then a factor on the LHS and RHS of the mass matrices in the algorithm. At the end, these multiplications double the number of repetitions to reach 8 times for σ_2, and 16 times for σ_4. It it important to notice that the sixth joint does not appear explicitly in the LFT model at hand, because the effector is model as a point mass at the end of the sixth link. Hence, a motion along this axis does not change the manipulator inertia mass matrix since the sixth joint turns around its axis of symmetry.

This uncertain representation of the mass matrix is then inverted to define the linearized model on which the μ-analysis is led, based on the structure described in Figure 8.10b. The controller synthesized in the previous section is used to close the lower loop. From the Lyapunov theory, the closed-loop system is supposed to stay stable whatever the gain setting, provided that the gain matrices are positive-definite. The analysis of stability is thus performed to assess the numerical conditioning of the space robot problem w.r.t. the SMAC

toolbox tools. It clearly appears from the following results that the problem is hard to solve by μ-analysis, since the bounds obtained are rather conservative. New methods are also presented to overcome this conservatism in (Roos et al., 2011), but prove to be inefficient on the present problem because no lower bounds were available. The advantage of these tools though is their extension to the robust performance in the next section.

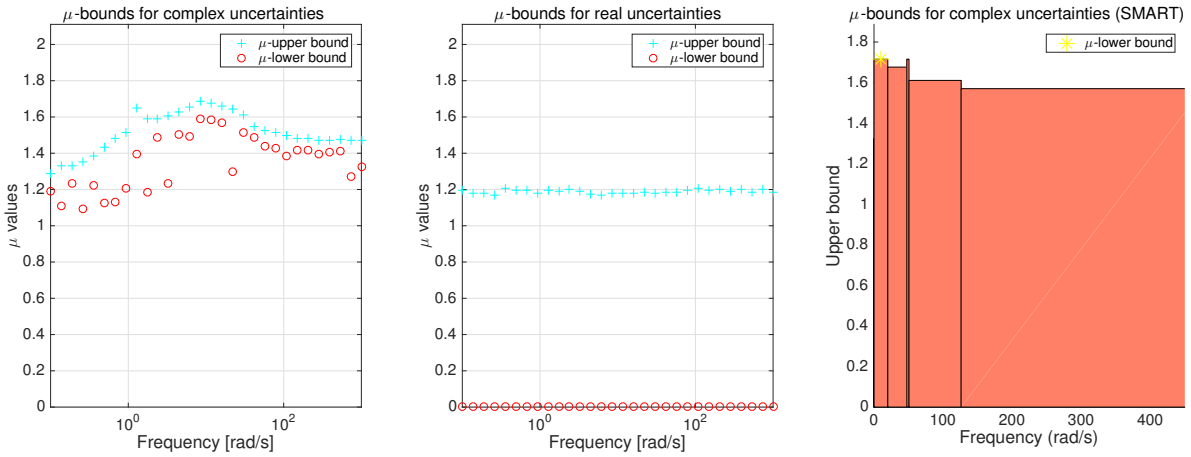
The μ-analysis is based on the computation of the maximum norm of the uncertainty before the loss of stability or performance (Alazard et al., 1999). This problem is reduced to the computation of a structured singular value, denoted μ, which is almost impossible to solve analytically. Hence, methods have been developed to derive upper and lower bounds around this value instead. If the algorithm yields a small enough gap between these bounds, then the robust stability or performance is assessed according to the approximate value obtained for μ. The bounds are derived based on a given range of variation of the parameters. The computation of these bounds through the μ-analysis allows to compute the subset of the parameters that guarantees the stability of the uncertain system. Since the varying parameters are normalized in the following developments, the stability will be reached if the upper bound is at least lower than 1. For example, if only one joint angle was considered varying in an initial range of $\sigma_2 \in [a, b]$, the following reduced subset guarantees the stability in a conservative way:

$$\sigma_2 \in \left[\frac{a}{\mu_u}, \frac{b}{\mu_u}\right]$$

where μ_u denotes the upper bound of μ, defining then a more conservative range for the uncertainty. If this value goes under 1, then the robust stability would be ensured for the initial set of variation. Otherwise, the lower bound must be obtained to assess the conservatism of the upper one. In the present case, the lower bound is 0 since the system is always stable, as seen in the previous section. Nevertheless, the numerical upper bound obtained by the SMART toolbox is rarely smaller than one, and lead then to very conservative subsets of stability. They are derived in the following for both parameterizations of the rotation and use the rigid model of the space robot. Indeed, the flexible case proves to be too cumbersome to be numerically tractable by the SMART tools.

Numerical results for both parameterizations The bounds obtained for the two parameterizations in $\sigma_2 = \tan(\theta/2)$ and $\sigma_4 = \tan(\theta/4)$ are summarized in Table 8.5. In this table, the initial ranges of variation of the joint angles are presented at the line "Limits" and come by design according to the specifications given by MDA.

One major remark is that the μ algorithms are numerically very sensitive to the size of the

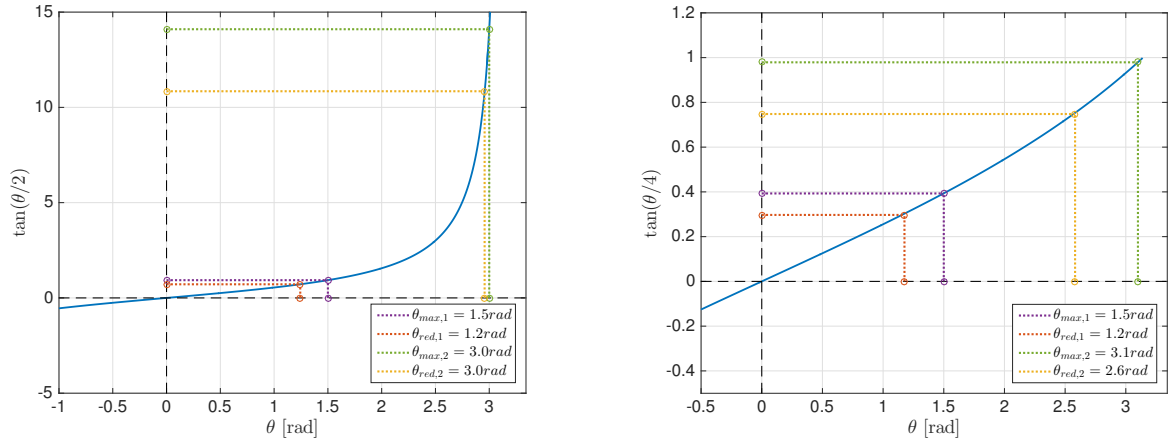


(a) (b) (c)

Figure 8.14: Computation of the μ-bounds for stability with σ_2-parameterization; (a) Native `Matlab` function `mu` for complex uncertainties; (b) Native `Matlab` function `mu` for real uncertainties; (c) SMART function `muub/mulb` for complex uncertainties.

global uncertain block $\mathbf{\Delta}_\sigma$. This is why the second parameterization, which leads to a block twice the size of the first one, yields poorer results on the upper bound. The second point is about the inclusion of the real or complex nature of the uncertainty. It is well known that the algorithms are much efficient and stable to derive the bounds for a complex sub-block than for real ones, knowing that one real uncertainty is the most difficult case (Alazard et al., 1999). To this end, the SMART toolbox integrates dedicated functions for the computation of the lower bound with real uncertainties, namely `mulb_nreal` for any number of real variables, and `mulb_1real` for a unique real parameter (Roos, 2013). The results are presented for both complex and real cases in Figure 8.14.

A last remark is made on the resulting ranges of variation obtained from the μ-analysis. Since the tangent function is used to parameterize the joint angle, the conservatism on the upper bound has tremendous effects on the resulting stability subset. As shown in Figure 8.15, depending on the absolute value of the upper limit on the joint angle, a same μ-bound can reduce the subset drastically when the initial value was low. This result is due to the tangent function shape, with very narrow variation close to the origin. This parameterization leads then to great cuts on the joint angles that were initially small, while the wider ranges of variation are less affected, as seen in Table 8.5.



(a) (b)

Figure 8.15: Reduction of the joint upper limit by a μ-bound of 1.3; (a) Impact for the σ_2-parameterization; (b) Impact for the σ_4-parameterization.

As a conclusion, the values obtained in Table 8.5 show that the σ_2-parameterization leads to smaller bounds, meaning that the joint angles can vary in a wider range before destabilizing the system. Due to the higher size of its block Δ_σ, the σ_4-parameterization yields more conservative results by more narrow ranges of variation.

Further results for a reduced number of varying joints The stability margins are now derived from a different point of view. An iterative process is used by adding a new varying joint angle at each step. The process starts the analysis with only one varying parameter, derives the corresponding μ-bounds and reduces accordingly the range of variation to reach the stability. Then, an additional uncertainty is added to represent the next joint angle and new bounds are derived for this augmented block $\mathbf{\Delta}_\sigma$, until the six joints of the arm are included. It is worth reminding that the sixth joint angle does not appear explicitly in the LFT model at hand, such that the process either starts with the first or the fifth angle. The numerical results are summarized in Table 8.5, where the bottom-up approach "1-5" starts with the first angle and goes outward, and the top-down approach starts with the fifth angle and goes inward. Since the bounds are computed at each step, the final joint ranges are obtained by different reduction factors $1/\mu_{u,i}$, which differs from one joint to the other. Therefore, no values are reported in the table for these approaches since they are not global but specific to each joint.

The first approach yields wider ranges of variations for the last angles, while the second one does the opposite. Since the most important segments for the global motion of the arm are

Table 8.5: Subsets of the joint angles guaranteeing stability by μ-analysis

Case	Value	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	$\mu_{\mathbb{C}}$	$\mu_{\mathbb{R}}$
Limits	Min	-90°	-45°	0°	-180°	-90°	-180°		
	Max	+90°	+180°	+180°	+45°	+180°	+180°		
σ_2-Stability	Min	-72°	+141°	+145°	-174°	+134°	-175°	1.72	0
	Max	+72°	+174°	+174°	-141°	+174°	+175°	1.72	1.39
σ_4-Stability	Min	-24°	+56°	+82°	-115°	+27°	-180°	5.85	0
	Max	+24°	+115°	+127°	-56°	+100°	+180°	5.85	3.97
σ_2-Iterative 1-5	Min	-53°	+159°	+156°	-174°	+89°	-175°	-	-
	Max	+53°	+173°	+174°	-140°	+175°	+175°	-	-
σ_2-Iterative 5-1	Min	-88°	+132°	+148°	-174°	+150°	-175°	-	-
	Max	+88°	+174°	+174°	-154°	+174°	+175°	-	-
σ_4-Iterative 1-5	Min	-9°	+65°	+88°	-146°	-40°	-180°	-	-
	Max	+9°	+108°	+124°	-14°	+150°	+180°	-	-
σ_4-Iterative 5-1	Min	-60°	+56°	+94°	-101°	+47°	-180°	-	-
	Max	+60°	+115°	+118°	-72°	+83°	+180°	-	-

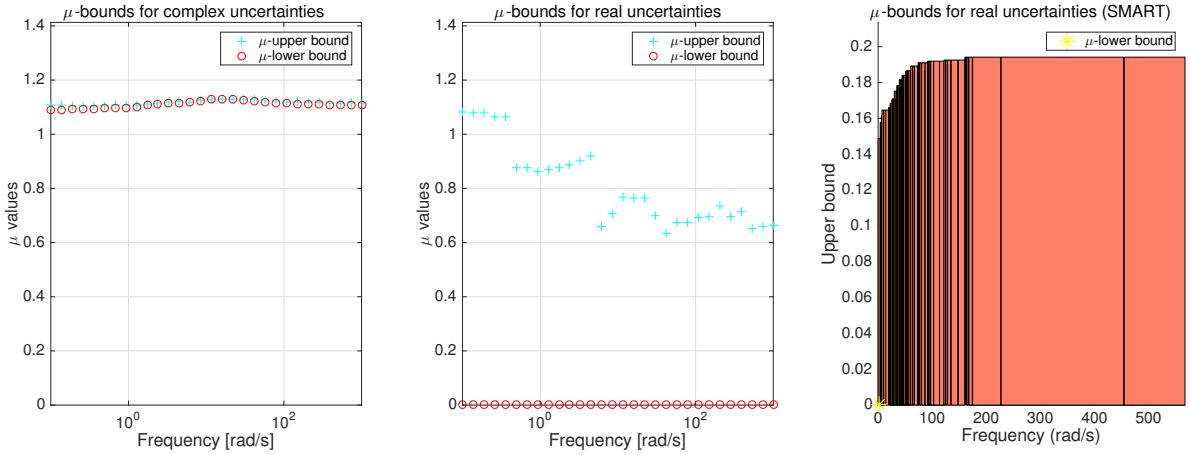
the first three, one may conclude that the top-down approach is advantageous. Indeed, the first joint has much influence on the dynamics and, as a consequence, its bounds are more conservative. When starting at the bottom of the arm, all the following joint ranges are impacted by this conservatism. On the other hand, starting at the last segment, which has less influence on the dynamics, the results become more conservative only at the last steps.

A last comment can be made on the μ-analysis with the single parameter σ_{2_5} at the first step of the process. For these computations, the SMART tools prove to perform better than the native `Matlab` function `mu`, as illustrated in Figure 8.16. Indeed, the upper bounds is almost zero and provides thus the robust stability for the whole range of variation of the fifth joint angle, while the others are fixed in a nominal configuration.

8.3.4 Robust performance analysis

The μ-analysis is also known to be a powerful tool to study the robust performances of linear systems (Alazard et al., 1999). The main requirements to perform this analysis is to first reduce the subset of the uncertainties to ensure the robust stability of the system. Then, the given performance transfer is fed back through an additional complex block $\mathbf{\Delta}_p$ to yield the augmented uncertain block given in Figure 8.17. The same notation ($\mathbf{w}$, $\mathbf{u}$) is used to denote the performance transfer introduced in Figure 8.4.

The results are mainly presented for the rigid case, and only a brief word is said on the flexible performance using the `Matlab` functions `sigma` to evaluate the singular values of an



(a) (b) (c)

Figure 8.16: Reduced computation of the μ-bounds for stability with σ_2-parameterization of the fifth angle θ_5 only; (a) Native `Matlab` function `mu` for complex uncertainties; (b) Native `Matlab` function `mu` for real uncertainties; (c) SMART function `muub_mixed/mulb_1real` for real uncertainties.

uncertain model.

μ-analysis on the rigid case The previous results are extended to the performance transfer by adding the feedback illustrated in Figure 8.17. New functions must be used to derived the μ-bounds since the block $\mathbf{\Delta}$ is now mixing real and complex uncertainties. The upper bound is still obtained by the function `muub_mixed` while the lower one uses the function `mulb_mixed` instead of the previous `mulb_nreal` or `mulb_1real` for one parameter. The obtained bounds and the corresponding intervals of guaranteed performances are presented in Table 8.6. Similar to the robust analysis, the bounds are also plotted in Figure 8.18 using both the `Matlab` function `mu` and is compared to the results of the SMART tools.

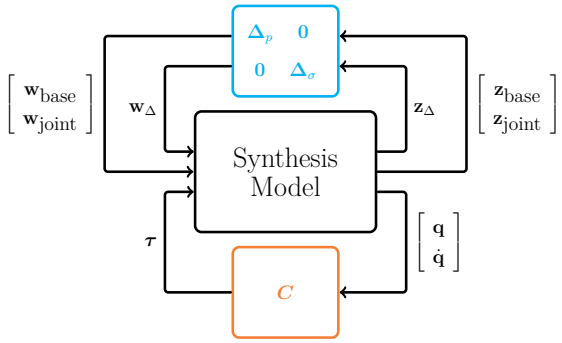
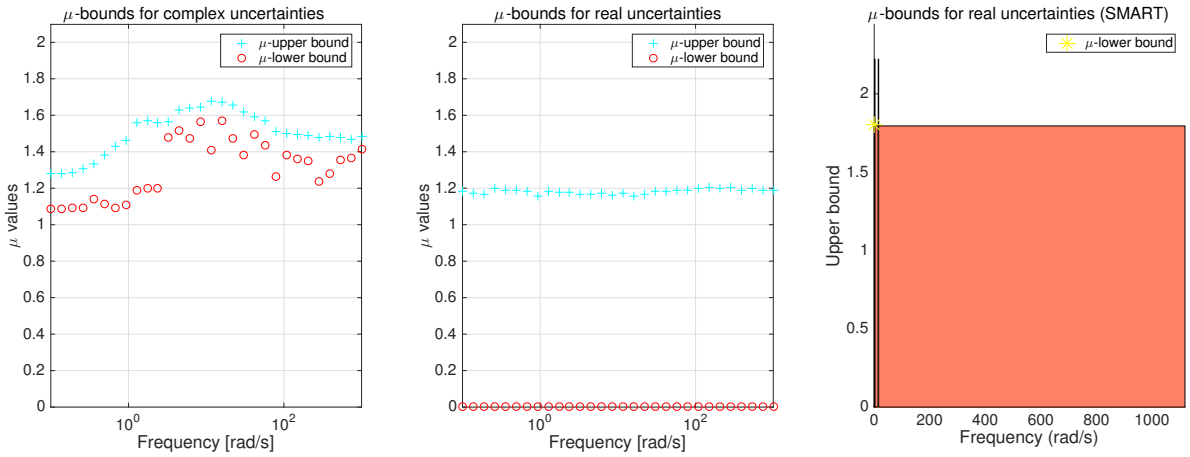


Figure 8.17: LFT form of the linearized model of the robotic system including the performance transfer.

The same conclusions hold on the robust performance analysis, namely that the σ_2-parameterization is less conservative and yields greater ranges of variation. As expected, the subset guarantee-



(a) (b) (c)

Figure 8.18: Computation of the μ-bounds for performance with σ_2-parameterization; (a) Native `Matlab` function `mu` for complex uncertainties; (b) Native `Matlab` function `mu` for real uncertainties; (c) SMART function `muub_mixed/mulb_nreal` for real uncertainties.

ing the robust stability are further shrunk to ensure the robust performance. The iterative approach is also more efficient on the robust performance problem, and provides the guarantee on a much wider range of variation for the first segment. At the end, the range of motion allowed by the μ-analysis is rather small and reduces the effector workspace to a limited cartesian space. The advantage of the iterative approach is to maximize the variation of the first joint, extending greatly the cartesian workspace.

Table 8.6: Subsets of the joint angles guaranteeing performance by μ-analysis

Case	Value	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	$\mu_{\mathbb{C}}$	$\mu_{\mathbb{R}}$
Limits	Min	-90°	-45°	0°	-175°	-90°	-175°		
	Max	+90°	+175°	+175°	+45°	+175°	+175°		
σ_2-Performance	Min	-49°	+161°	+162°	-173°	+160°	-175°	2.41	1.81
	Max	+49°	+173°	+173°	-161°	+173°	+175°	2.41	2.22
σ_4-Performance	Min	-22°	+58°	+84°	-114°	+29°	-180°	6.51	2.15
	Max	+22°	+114°	+127°	-58°	+99°	+180°	6.51	4.23
σ_2-Iterative 5-1	Min	-82°	+138°	+154°	-174°	+164°	-175°	-	-
	Max	+82°	+174°	+174°	-146°	+172°	+175°	-	-
σ_4-Iterative 5-1	Min	-58°	+56°	+92°	-100°	+58°	-180°	-	-
	Max	+58°	+116°	+119°	-74°	+72°	+180°	-	-

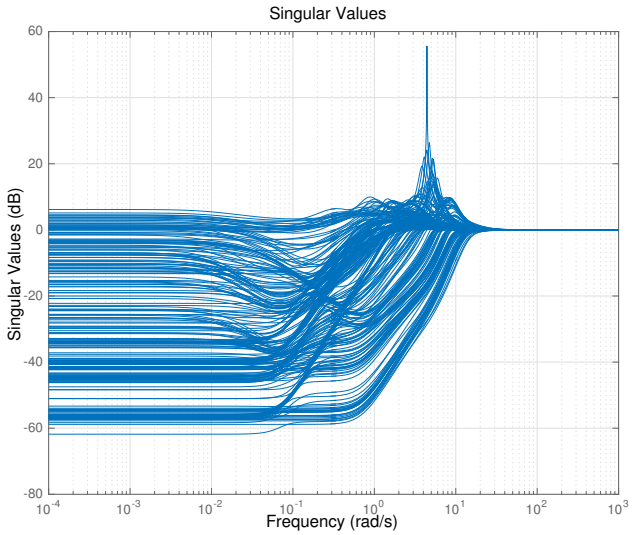


Figure 8.19: Singular values of the weighted performance transfer for the uncertain flexible model

μ-analysis on the flexible case Finally, the robust performance problem may be investigated by the `Matlab` function `sigma` to compute the singular values of a great number of randoms configurations. An illustrative example is provided in Figure 8.19, where the performance transfer between $(\mathbf{w}, \mathbf{z})$ in Figure 8.17 is plotted for random configurations among the ranges of variation. It turns out that the maximum value for these random cases is at least a lower bound of the real μ-bound. In the example, a value of 55.6 dB is obtained, which corresponds to a factor of reduction of 99.8% on the uncertain parameters to reach the robust performance.

Is is obvious that the simple PD controller performance is limited on the flexible case, and should be augmented with a roll-off contribution of even by a wide notch filter on the frequencies of the peaks.

Conclusion

The control of a space robot has been investigated throughout this section. The control architecture was first described with the two different loops that control the open-loop behaviors of the system and its guidance at a higher level. A fast control based on PD controllers was synthesized to ensure a given requirement on the disturbance rejection, while the second loop was based on a partitioning of the effector reference trajectory between the base and the manipulator motions. These control laws were tested in simulation in Section 9.1 to asses the performances and the limitations of the present design.

CHAPTER 9 VALIDATION OF CONTROL LAWS

The validation of the safe capture of a tumbling debris by a space robot is detailed in this last part. Many steps have already been validated separately. The dynamics simulation has been checked, based on the energy drift along time, and it was shown how some of the approximate models preserved this quantity while reducing the simulation time. From the control point of view, a robust analysis has been performed on the closed-loop system including the flexible behavior of the segments. Lastly, the guidance of the arm has been validated by numerical considerations, ensuring that the trajectory can track the target, and eventually synchronizes exactly with it. In the sequel, the global validation is led by interconnecting all these sub-parts together. A numerical scheme is introduced first, and then the robotic test bench used at TAS is detailed along with the results obtained experimentally.

The numerical scheme used in simulation is introduced first. All the previous parts of the thesis are summarized into this scheme: the dynamics chapter is used to simulate the behavior of a space robot in weightlessness, the control one provides a closed-loop controller coping with flexibilities and performances, and the path planner generates smooth trajectories of capture for the robot guidance. A global scheme is then introduced and the numerical results of simulation are discussed in the following.

Finally, a second part addresses more specifically the problem of validating space dynamics or kinematics with Earth-based system, like the robotic test bench used at TAS. This crucial issue is first discussed to highlight the outcomes of performing terrestrial experiments for space systems. The test bench is quickly detailed through the physical components available and the network used to ensure the real-time coupling between simulation dynamics and the real robots motion and measurements. The few results obtained are introduced and discussed at the end.

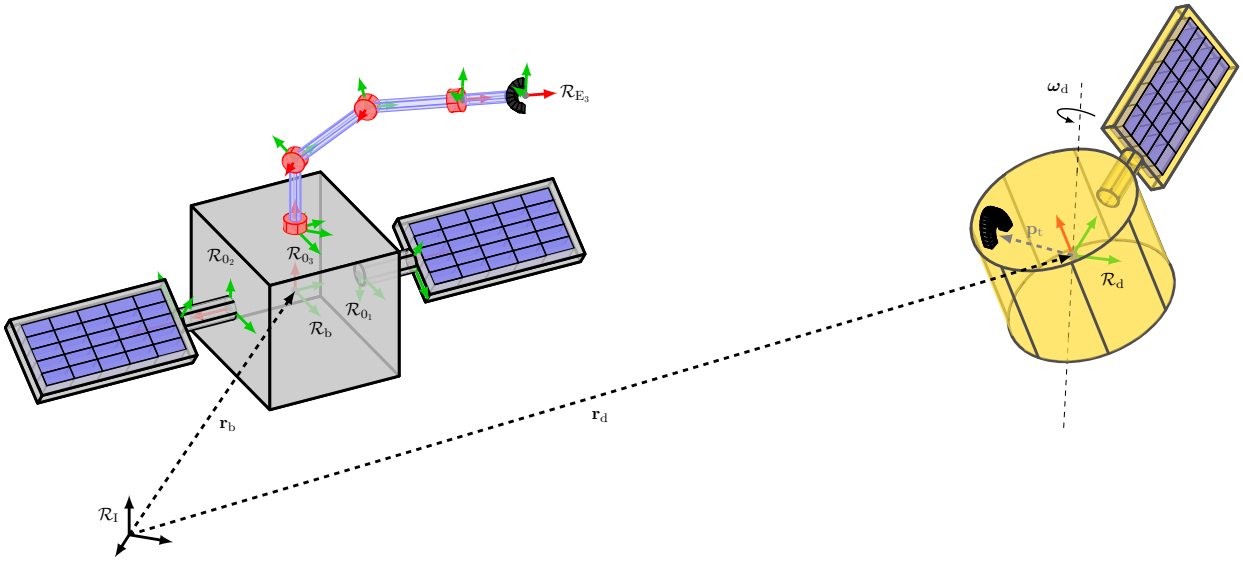


Figure 9.1: Schematic view of the bodies and frames involved in the simulation

9.1 Numerical Validation

The numerical validation of the control laws is performed based on the architecture introduced in Figures 8.1 and 8.2 for the kinematic scheme recalled in Figure 9.1. The global control scheme is built using the software `Matlab/Simulink` and the sub-blocks are coming from the different parts of the thesis, as illustrated in Figure 9.2. Its structure is first described to emphasize the design features of each sub-blocks, and then the numerical results of capture are presented. The discussion is led for the centralized and decentralized controllers of the manipulator, as well as for the rigid and flexible models of the whole spacecraft.

9.1.1 High-fidelity simulator

The elements of the simulator presented in Figure 9.2 are detailed hereafter w.r.t. their goal in the simulation. Each of them represents one topic of the thesis, and is implemented in a general manner, in order to simulate the behavior of any space robot.

The following notation is used in the scheme:

`FrameExpression_x_FrameReference_Body`

where (`x`,`xDot`,`xDDot`) denote a state vector and its successive time derivatives, `Body` is the body considered, `FrameReference` is the frame of reference w.r.t. which the states are derived, and `FramExpression` is the frame of expression of the states. The frames are either

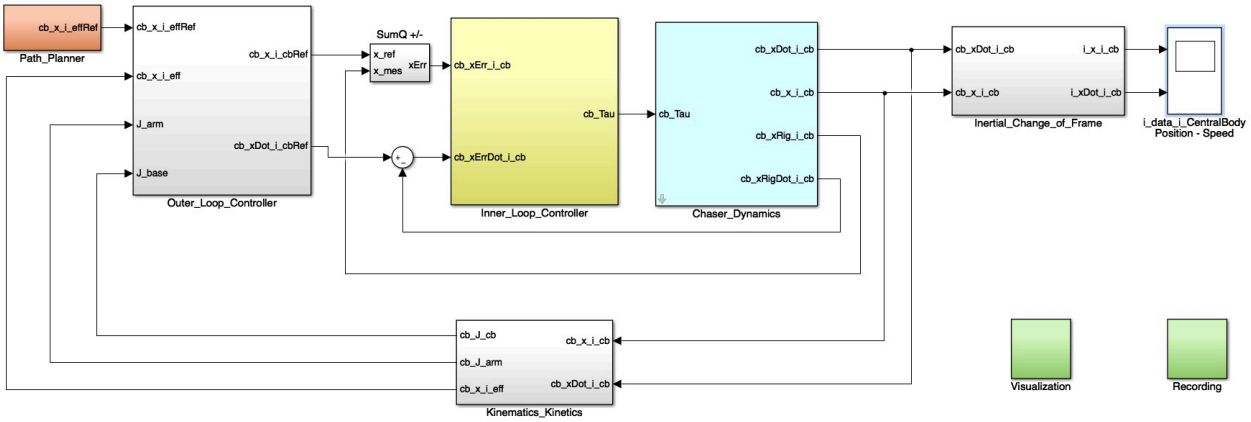


Figure 9.2: Simulation scheme based on the inner/outer loops

the inertial frame, denoted by "i", or the chaser body-fixed frame, denoted by `cb`, for "central body". Concerning the bodies, the names `cb` and `eff` are used to denote the quantities related to the central body and to the effector. The addition of `Ref` at the end of the signal names refers to the reference quantities used in control.

As seen in Figure 9.2, the main blocks of the simulator represent the global control architecture introduced in Figure 8.2. They are listed as follows:

Path planner This block contains the optimal capture trajectory introduced in Chapter 7, that serves as reference for the end-effector guidance. These data are recorded prior to the simulation, based on the propagation of the debris dynamics and of the resulting target trajectory. The output is the effector trajectory expressed in the chaser body-fixed frame;

Outer Loop The outer loop contains the guidance algorithm presented in Section 8.1. Based on the desired end-effector trajectory, the coordinated control invert the Jacobian matrix to obtain the reference to follow by the central body and by the manipulator joints;

Inner Loop The low-level loop consists in the PD controller designed and studied in Chapter 8. The decentralized structure between the AOCS and the arm controller is implemented inside this block;

Chaser Dynamics The whole modeling part of the thesis in Chapters 4 to 6 is summarized in this block. The inverse and forward dynamics are performed simultaneously to compute respectively, the nonlinear Coriolis and centrifugal terms, and the inverse of the mass ma-

trix. It applies for both rigid and flexible segments thanks to the common dynamic models introduced earlier. The resulting states accelerations are also integrated twice inside this block, to provide the inertial position and speed of the chaser expressed in its body-fixed frame;

Kinematics The flexible kinematics and kinetics provide the end-effector states, in both position/attitude and linear/angular speed. In addition, the Jacobian matrices are computed in this block to minimize the number of embedded scripts in the scheme;

Inertial Change of Frame This block performs only the rotation of the input quantities from the chaser body-fixed frame to the inertial frame;

Visualization/Recording These last blocks gather all the signals of the scheme to plot them and save them in the `Matlab` workspace.

This high-fidelity simulator is based on the flexible model of a space robot, according to the models derived at the beginning of the thesis. The data used for the spacecraft and for the manipulator are described in appendices D.1.3 and D.2.1. The spacecraft is based on the DEOS project and weighs around $800\,kg$, while the manipulator comes from a design performed by MDA for the French company TAS. It weighs $63\,kg$ and is 6-meters long with two main flexible segments. The modes obtained for this manipulator are provided in appendix D.1.3, and some of them exhibit low frequencies around $4\,rad/s$. Recalling that the manipulator controller is designed for a reference frequency of $0.7\,rad/s$, these modes create great difficulties in simulation due to the simplicity of the controller.

The results of these simulations are now presented to compare the decentralized and centralized controllers of the manipulator. They are thoroughly studied on the rigid model, but the closed-loop flexible one proves to be highly unstable in simulation.

9.1.2 Simulation results

The data used to assess the performances of a given controller are based on the list of requirements fixed in Table 8.1. They are summarized as follows:

- The capture trajectory must be tracked with a maximal error of $1\,cm$;
- The base must be maintained below the limit of $0.3°$ for communication purpose;
- The base actuators and the joint torques are subject to saturations.

Besides, the base consumption is also evaluated by integrating the base command (i.e., forces commanded to the thrusters) to compute the mass of fuel consumed during the maneuver.

The following relation is used to obtain it ((Weeden, 2011), p.531):

$$m_{fuel} = \frac{1}{g_0 \, I_{sp}} \int_0^t \|u(t)\| dt \tag{9.1}$$

with $g_0 = 9.81\, m/s^2$ denoting the Earth gravity acceleration, I_{sp} the thrusters specific impulse, and eventually $u = \boldsymbol{f}_\mathrm{b} + \boldsymbol{n}_\mathrm{b}/d$ the thrusters command according to the required AOCS force $\boldsymbol{f}_\mathrm{b}$ and torque $\boldsymbol{n}_\mathrm{b}$. The distance d stands for the lever arm from the thrusters location to the base CoM.

9.1.2.1 Rigid case

The results are presented for the rigid model of the space robot with the decentralized controller in Figure 9.3, and with the centralized one in Figure 9.4. First, it clearly appears that the capture requirement is fulfilled: **both controllers synchronize the end-effector with the desired accuracy of 1 cm**. There is a slight advantage for the centralized controller, whose the error remains the lowest. The main difference lies in the base reaction to the manipulator motion.

Indeed, the AOCS controller is not sufficiently effective to reject the disturbances coming from the arm. This drawback is due to the slow requirement used on its weighting function during the synthesis. A large difference is noticeable on the base motion, that prevent the fulfillment of the attitude requirement of 0.3°. The maximum angular drift is around 1° in the centralized case, and it rises to 4° in the decentralized one.

Having a look on the resulting base torques, this result suggests that the actuators on the base are not capable of managing the arm disturbances. Indeed, the peaks of command are always crossing the limits of $5\, Nm$. One may think of using CMG instead of *Reaction Wheel (RW)* for example. It is also worth mentioning that the Jacobian inversion in the guidance may be the source of these peaks. They are observed on the joint torques and are then transmitted to the base. Since no feedforward control is considered in this controller, the base does not anticipate from the discontinuous motions of the manipulator.

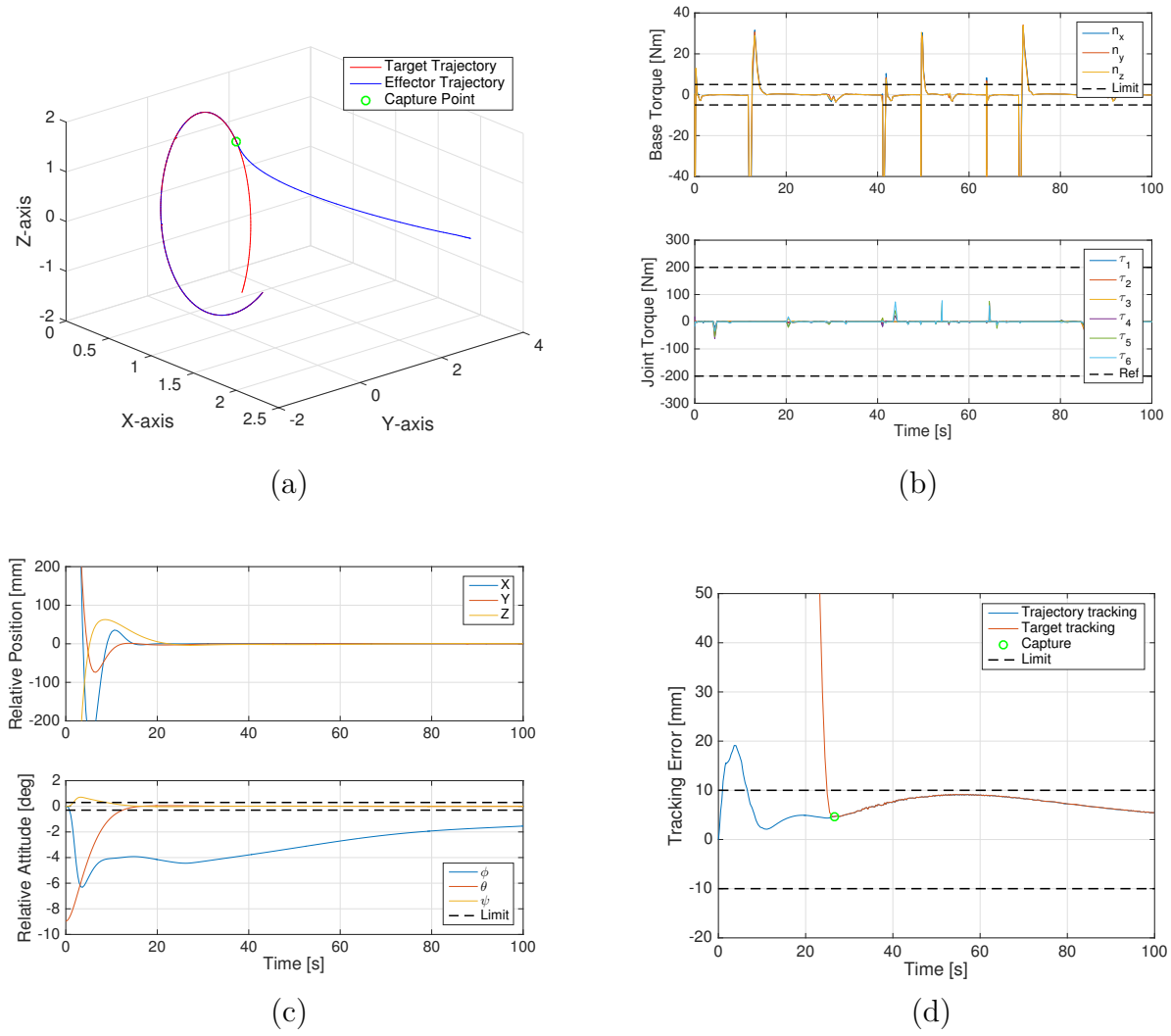


Figure 9.3: Results of simulation for the decentralized controller on the rigid space robot model; (a) 3D trajectory of the end-effector and of the target; (b) AOCS efforts and joint torques; (c) Linear and angular drift of the base during the maneuver; (d) Norm of the tracking error of the end-effector w.r.t. the desired capture trajectory, and w.r.t. the real target position.

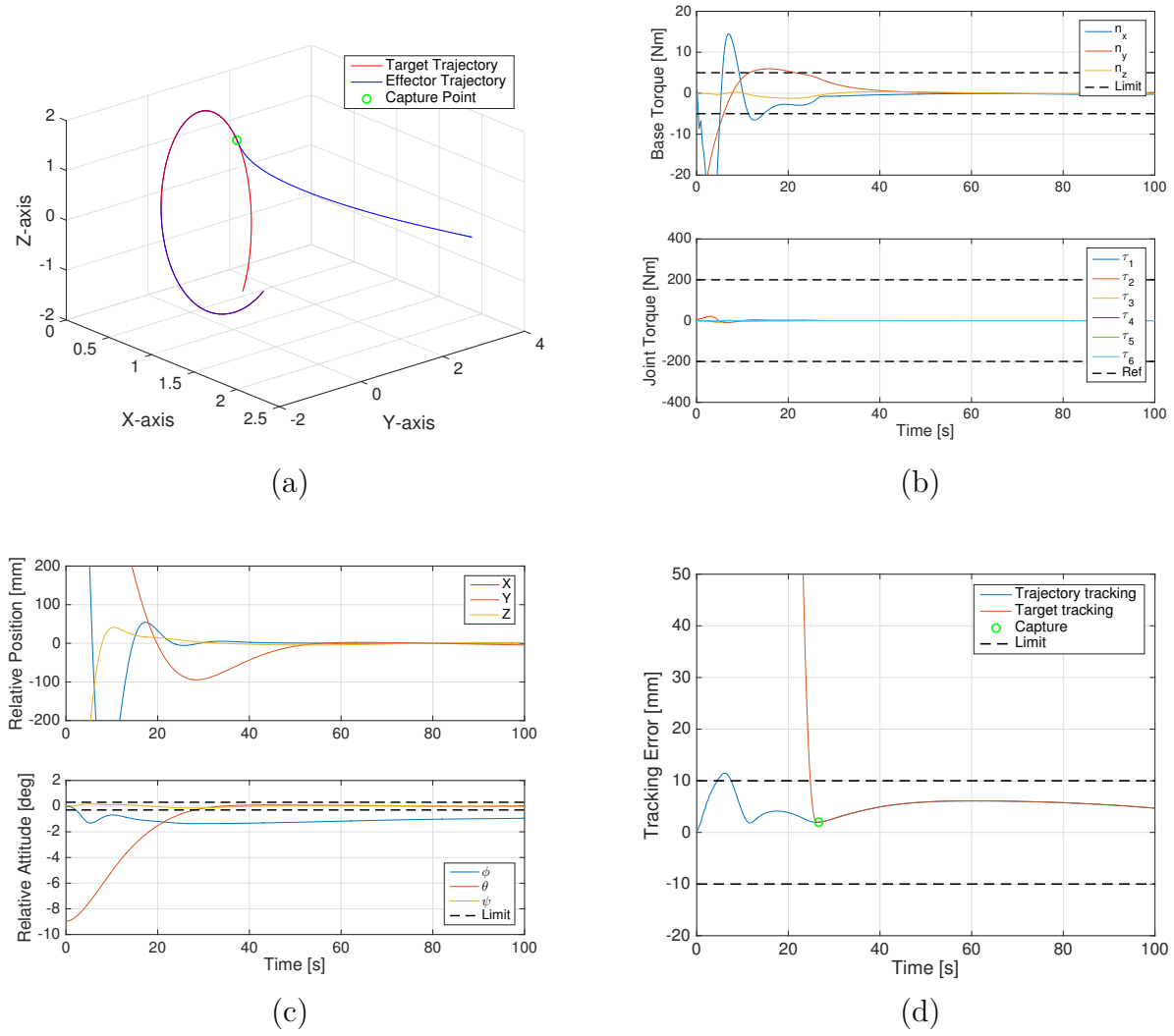


Figure 9.4: Results of simulation for the centralized controller on the rigid space robot model; (a) 3D trajectory of the end-effector and of the target; (b) AOCS efforts and joint torques; (c) Linear and angular drift of the base during the maneuver; (d) Norm of the tracking error of the end-effector w.r.t. the desired capture trajectory, and w.r.t. the real target position.

A last analysis is performed on the mass of fuel used during the maneuver. The consumption of both strategies are presented in Figure 9.5. On this graph, the centralized controller reduces drastically the consumption of the base actuators by minimizing the disturbances coming from the arm. The AOCS torques mainly exceed the limit during the initial base maneuver to reach the safety position. After that, they remain inside the bounds. This conclusion is further supported by the base torques in Figure 9.4b, and by the diagonal terms of the proportional gains gathered for the AOCS and the manipulator in Table 9.1. Indeed, the centralized controller accounts for the arm coupling and, as a consequence, its matrices contains smaller gains since they are full. On the other hand, the decentralized controller results in higher gains only on the diagonal, that make it more sensitive to the noise introduced by the Jacobian inversion.

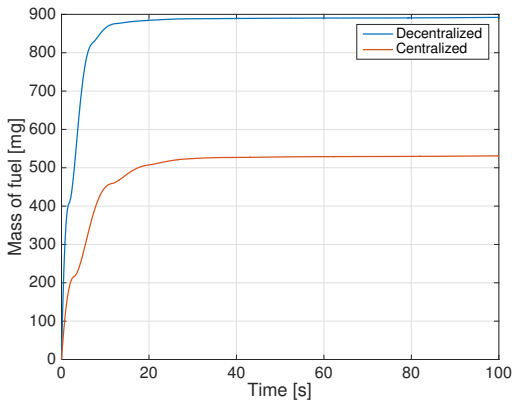


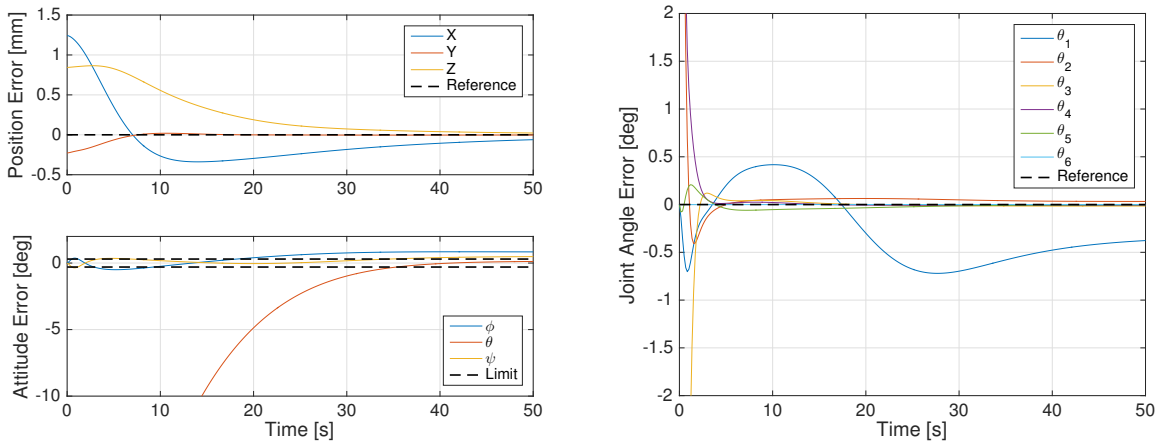
Figure 9.5: Mass of fuel consumed during the maneuver of capture to maintain the chaser at a safe distance and to counteract the disturbances coming from the manipulator motion

The main conclusion of these analyses is that the guidance seems to be the main source of concern for the compliance with the set of requirements listed above. Indeed, when the arm is close to a singular configuration, the inversion of the ill-conditioned Jacobian matrix leads to discontinuous references for the inner loop. This phenomenon creates the peaks observed on the joint torques, and their transmission to the base causes a higher fuel consumption. A more robust guidance scheme should be developed to better handle the singular configurations.

An alternative solution is to use the centralized controller with smaller gains, in order to limit the sensitivity to the numerical noise. As illustrated for this case, the command profile on both the manipulator and the base remains smooth and reduces drastically the fuel consumption in the absence of singular configurations.

Table 9.1: Proportional gains setting of the decentralized and centralized arm controllers

Base	Axis	x	y	z	ϕ	θ	ψ
	diag($\boldsymbol{K}_{p_{dec}}$)	58	286	469	23	1539	1668
	diag($\boldsymbol{K}_{p_{cen}}$)	18	19	170	35	243	135
Arm	Joint	1	2	3	4	5	6
	diag($\boldsymbol{K}_{p_{dec}}$)	157	471	198	84	76	0.002
	diag($\boldsymbol{K}_{p_{cen}}$)	46	518	158	642	7	0.003



(a) (b)

Figure 9.6: Results of open-loop simulation for the decentralized controller on the flexible space robot model; (a) Errors on the relative coordinates of the base w.r.t. a given reference; (b) Errors on the joint angles w.r.t. a given configuration.

Flexible case The flexible case proves to be very unstable in simulation. Only the inner-loop remains stable when a reference is set on the base coordinates and on the joint angles. Otherwise, the outer-loop is very sensitive and becomes easily unstable when the current guidance scheme is tuned manually. A synthesis method should be developed to ensure that the global loop is stable. There are two main sources of instability: the numerical ones, and the intrinsic ones. The first ones are assumed to be handled automatically by the advanced solver embedded in `Simulink`, or by setting manually a fixed time step, small enough to prevent any unstable behavior.

On the other hand, the intrinsic stability of the outer loop should be studied more thoroughly to set the gains of the control law, G_b and G_m. An open-loop test is presented to assess the inner-loop performances in Figure 9.6. A step reference is sent on each input: the base position and attitude, and the joint angles. It results that the PD controller is very limited on the flexible model, even with very high gains. The time response on each input is around $50\,s.$, which is too slow to prevent the unstable behavior in closed loop. Therefore, a more sophisticated controller should be designed on the flexible model to provide a stable simulation of capture.

9.2 Experimental Validation

Experiments have been conducted from the fall of 2014 to the spring of 2015 at TAS. The goal was to perform the simulation of a robotic capture with HIL. The test bench used for this experiment is illustrated in Figure 9.7. Two industrial robotic arms IRB 2400/16 made by ABB were used to simulate the behavior of the target from one side, and of the chaser on the other. This latter was mounted on a base moving on a rail of 10 meters, allowing to reproduce the phase of rendezvous as well. In the sequel, a quick discussion is led about the validity and the added value of terrestrial experiments for space system like the space robot, then the components of the test bench itself are described, and finally the few results obtained are presented.

Figure 9.7: Robotic test bench used to simulate the on-orbit kinematics of a capture

9.2.1 Pros/Cons of an Earth-based Simulator

Keeping in mind that the robotic motions of the test bench aims at reproducing the behavior in weightlessness of a tumbling debris and of a moving-base manipulator, inherent limitations exist due to the gravity field. The main problem is to faithfully mimic the free-floating dynamics while counteracting this gravity, which obviously reduce the potential validations in terms of kinematics, kinetics and dynamics.

From a dynamic viewpoint, it seems quite clear from the literature review that conditions of weightlessness cannot be reproduced on Earth because of the gravity field. The best compensation systems, like the underwater facilities used to train the astronauts, are always introducing bias efforts that do not exist on orbit, like friction and drag. Concerning the

gravity compensation based on springs or tethers, they only compensate the weight in one direction and often in a static case of equilibrium, not in a dynamic one. The best compromise between the costs and the faithful behavior in weightlessness seems to be the planar test bench with air bearings. The 2D dynamics are almost exactly reproduced since the friction can be lowered and large areas of motion can be used (Rybus et al., 2013). Therefore, using the TAS robotics test bench base on industrial manipulators, it appears clearly that only kinematics and kinetics validation are available.

Indeed, one of the trendy axis of research nowadays is the HIL validation of space components like cameras or lasers (Boge et al., 2010). The idea is to simulate in real-time the dynamics in weightlessness, to reproduce the resulting kinematics using a fast control loop on the industrial manipulators that mimic the spacecrafts, and then to check the behavior of the component undergoing test with mockups and illumination conditions (Aghili et al., 2010; Xu et al., 2007). One major limitation in this type of experiments is the time delay introduced during the real-time simulation, since it can destabilize the global closed-loop system by further slowing down the feedback of the component itself. This latter is already costly in terms of time with advanced image processing for the camera for example. This bias would even prevent an effective component from being validated or from reaching its best performances, if the additional time delay of simulation is too important. Thus, the dynamics simulation must be optimized and the communication delays need to be well understood prior to any HIL experiment. One last problem of HIL is the need for a realistic environment. That means complete darkness in background, and specific directional lighting conditions to mimic the Sun and test the image processing performances in illumination conditions. Many of these requirements are fulfilled on the DLR's robotic test bench called *European Proximity Operations Simulator (EPOS)*, which could serve as a reference setup for further improvements.

Once these limitations are taken into account, a robotic test bench with industrial manipulators offers many ways to investigate the control stability and performances with actual sensors in the loop. The main advantage of this approach is to introduce the real noise and delay of the components, and to check their compatibility with the requirements specification. Using only numerical simulation with a network of computers, the costs are greatly reduced compared to on-orbit experiments or zero-gravity test beds (Stoll et al., 2012). For the example of the TAS test bench, the real on-orbit dynamics is reproduced by a high-fidelity simulator and a *Bumblebee* camera is embedded on the chaser robot to provide a relative position and attitude feedback of the target. This setup allowed to validate capture devices as well, using prototype grasping tools mounted on the manipulator dedicated to the chaser.

An illustrative example of this kind of validation is given by the CSA test bench in (Rekleitis et al., 2007).

The TAS test bench is now described into further details, and the results obtained for the simulation of a capture in weightlessness are presented.

9.2.2 Test bench layout

The test bench is made of two ABB manipulators. One is fixed on the ground and reproduces the behavior of the target, while the other is mounted on a rail and represents the chaser in approach. For sake of brevity, the industrial manipulators and their corresponding quantities are directly referred to as "chaser" or "target". First, the control architecture of each robot is presented based on their controller, called IRC5. Then the frames used in simulation are introduced. They allow to reproduce the motion obtained by simulation on the real test bench by applying some scaling and offset factors, in order to respect the manipulators workspaces. Then the simulator embedded on the real-time module is presented, to emphasize the additional blocks w.r.t. the previous simulation scheme.

Control network of the industrial manipulators The whole network structure is illustrated in Figure 9.8. Both robots are controlled by a dedicated IRC5 bay. PC3 is a computer embedded on the chaser robot for image processing, and eventually PC1 is the computer linked with the real-time module from dSpace. This latter uploads the real-time simulation scheme with the software Control Desk. The main steps of the experiment may be described as follows:

1. **PC1:** The `Simulink` scheme to emulate on-orbit dynamics is designed and the input/output signal of the experiments are chosen.
2. **PC1:** Using the dSpace release, the real-time application is built by `Matlab`.
3. **PC1:** The software Control Desk uploads and runs the real-time application onto the dSpace module.
4. **PC3:** While Control Desk is running the real-time simulation, a remote program initializes and monitors the communications between the simulation input/output and the different sub-systems, i.e., the robots and the camera. For both manipulators, the IRC5 bay manages the communication, receives the control inputs from the dSpace module, and sends it back the measurements. Regarding the image processing, an embedded C program runs during the simulation and sends constantly the relative distance from the target.

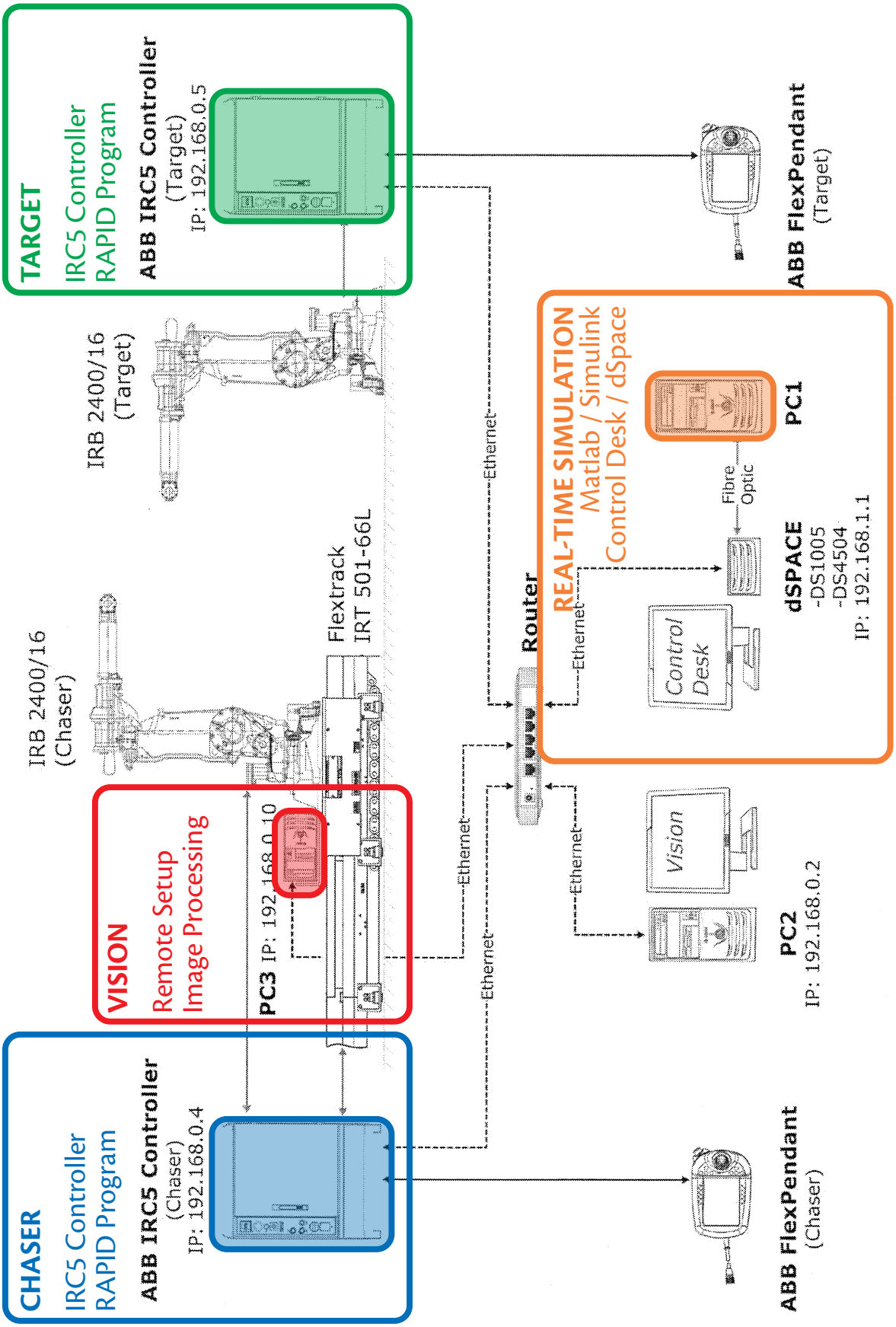


Figure 9.8: Robotic test bench setup to simulate the on-orbit kinematics of a capture

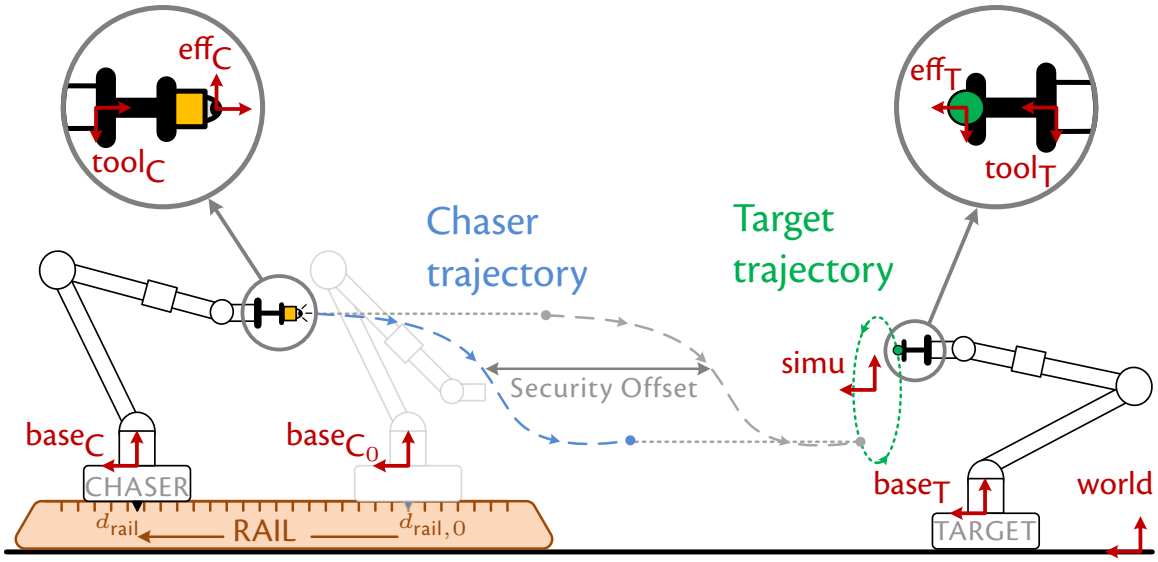


Figure 9.9: Frames introduced to describe the on-orbit capture on the robotic test bench

More precisely, each robot is managed by an IRC5 bay, containing all the necessary hardware to control the robot and to communicate with it using traditional protocols. It contains in its internal memory all the necessary key files to control the motion in the different frames defined by the user, and it is also used to define a security zone with a bounded workspace. Two solutions are available to control the manipulator: either by the remote panel linked to the IRC5 bay, called "FlexPendant", or remotely, using the communication network to communicate with this bay. Obviously, for the experimental validation proposed here, the manipulators are controlled using the ABB programming language, called "RAPID". A dedicated program manages the communication with the network in order to receive the commands, to perform them, and to finally send the current position and attitude of the tool back to the dSpace module.

Testbench frames In order to describe the motion of the robot in this environment, various frames have been introduced. They are illustrated in Figure 9.9, as follows:

- The **world** frame is attached to the room and is the main reference to express the distances from the observer point of view;
- The **simu** frame is the inertial frame of the real world environment. It is introduced to perform scaling and rotation w.r.t. the world frame, in order to make the spacecraft motion fit the actual workspace of the industrial robots;
- For each of them, three additional frames are introduced, denoted by the letter "C" to refer

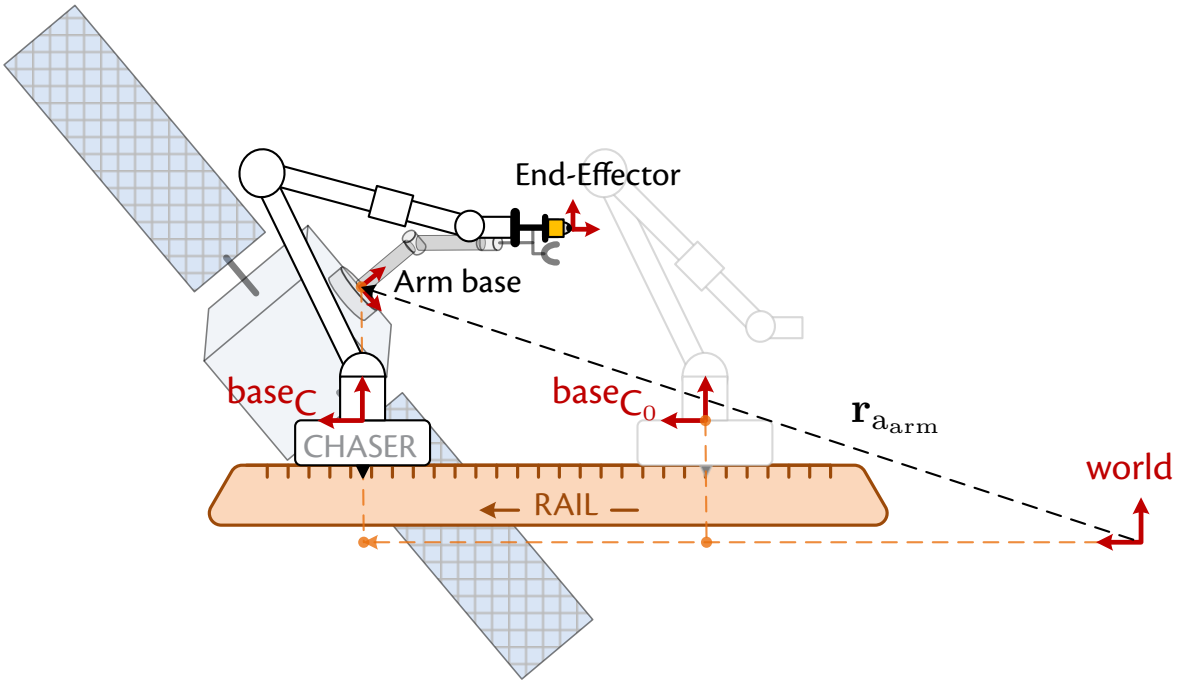


Figure 9.10: Rail motion according to the relative position of the arm base from the target in simulation

to the chaser, and by the the letter "T" for the target.

  - The **base** frame is attached to the manipulator base. For the target, this frame is fixed w.r.t. the world frame, but for the chaser, it is moving linearly along the rail axis;
  - The **tool** frame is attached at the end of the last segment, where the tool is mounted on the arm;
  - The **eff** frame is attached at the end of the tool. In the target case, it is located at the grapple fixture, which is the center of a green circle on the TAS bench. For the chaser, this point is located at the grasping mechanism, which is represented by the center of the camera lenses on the bench;

- One extra frame $\mathbf{base_{C_0}}$ is introduced to account for the motion of the chaser on the rail. As seen from Figure 9.9, this frame represents the reference position of the chaser on the rail, for which a calibrated distance has been computed w.r.t. the world frame.

Rail Motion The linear distance d_{rail} of the chaser is computed from this last frame base_C. The choice is made to reproduce the distance of the arm base from the target during the on-orbit motion. In other words, the relative position of the arm base w.r.t. the grapple fixture in the "real world" is projected along the direction of approach between both spacecraft, which is assumed constant. The position of the chaser on the rail is driven by this projection, which reduces the 3D motion to only one DoF. The parallel between the "real world" in simulation and its projection onto the test bench is illustrated in Figure 9.10.

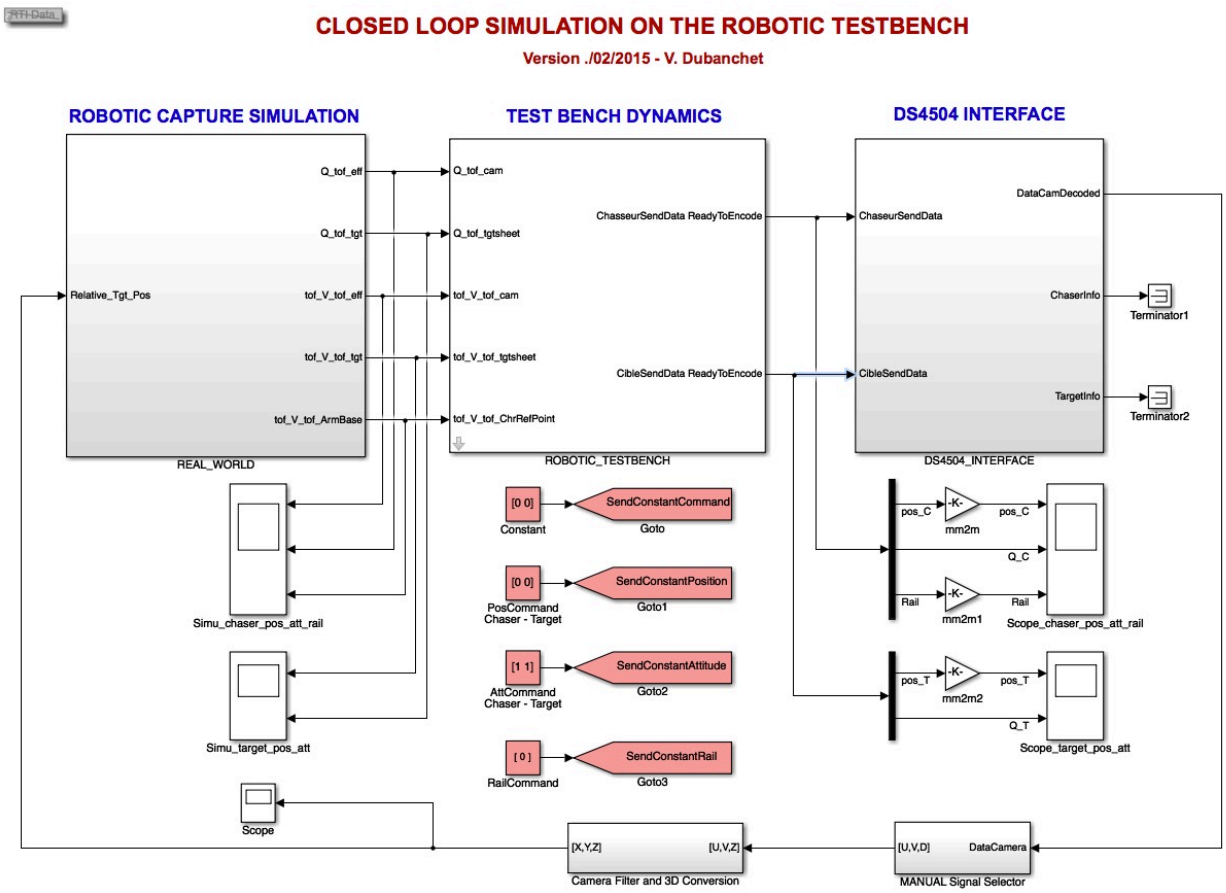


Figure 9.11: Global HIL structure of the `Simulink` scheme emulating the the on-orbit dynamics of the closed-loop system

Real-time simulator Regarding the simulator embedded on the dSpace module, a slight difference is introduced with the purely numerical one. Indeed, the experimental setup allowed to perform an HIL validation with the use of the Bumblebee camera and its dedicated image processing. Hence, the `Simulink` model include the feedback from a communication module receiving the data through the network and using them in the control of the chaser inside the simulation of the "real world". The numerical scheme of control for the whole test bench is illustrated in Figure 9.11.

Three main blocks are used to represent the different elements mentioned above. The first one, called "Robotic Capture Simulation", propagates the on-orbit dynamics of both the chaser and the debris using the previous scheme of the numerical validation step. The main difference is that the camera feedback is no longer computed from the simulation data, but comes from the real hardware components with the inherent noise and delay. Apart from that, the dynamics are simulated from the thesis rigid models for a single body, in the target

case, and for multi-body systems, in the chaser case. Indeed, the flexible models were not fully developed at the time of the experiments.

The second block, called "Test Bench Dynamics", contains the control strategy of the industrial manipulators. The on-orbit motions coming from the simulation are dispatched between the target and the chaser arms, according to the choices made for the **simu** frame introduced above. The block must provide "feasible" commands to the robots in terms of position and attitude, i.e., commands that lie in the workspace including the security limits. All the calibration data are also applied at this step, in order to reproduce as faithfully as possible the on-orbit kinematics. Since the robots are mounted on different supports, their axis of motion are not exactly aligned and these drifts must be accounted for when computing the commands. In addition, the commands sent to the IRC5 bays are modular and can be split w.r.t. to the linear and angular motion. For example, in the following results of simulation, the attitude of the end-effector embedded on the space robot is not reproduced because it would rotate the camera too. This would lead the target to go out of the camera field-of-view because of the implementation of a security offset, as illustrated in Figure 9.9. This is the only difference with the real on-orbit kinematics, except that the quantities from the simulation are scaled and translated to fit the test bench workspace.

Finally, the last block, called "DS4504 Interface", stands for the communication setup of the dSpace module. All the protocols and data encoding are implemented inside. They allow to write the messages in the form and with the tags required by the IRC5 bays and their RAPID programs, using the XML frames required by the ABB convention. Similarly, the measurements are received and decoded through this block, before entering the small camera block at the bottom of Figure 9.8. This block is actually an extension of the image processing step that was not yet implemented: it transforms the distances from pixels to meters. This is explained in more detail in the next sub-section.

Image processing The embedded camera on the chaser robot is a "Bumblebee2" stereo camera produced by PointGrey18. The image processing is performed on the PC3 computer on Figure 9.8, which is attached to the base of the chaser robot. Using the two images of its lenses, the depth information can be inferred from specific features in the picture. The resulting vector of position of the object in the picture is given in pixels in the plane of the camera, and in meters in the depth axis, as illustrated in Figure 9.12. Once computed, these data are sent to the dSpace module to be used in the `Simulink` model. Currently on the TAS bench, a sheet of paper with four green circles stands for the target. Each circle is ana-

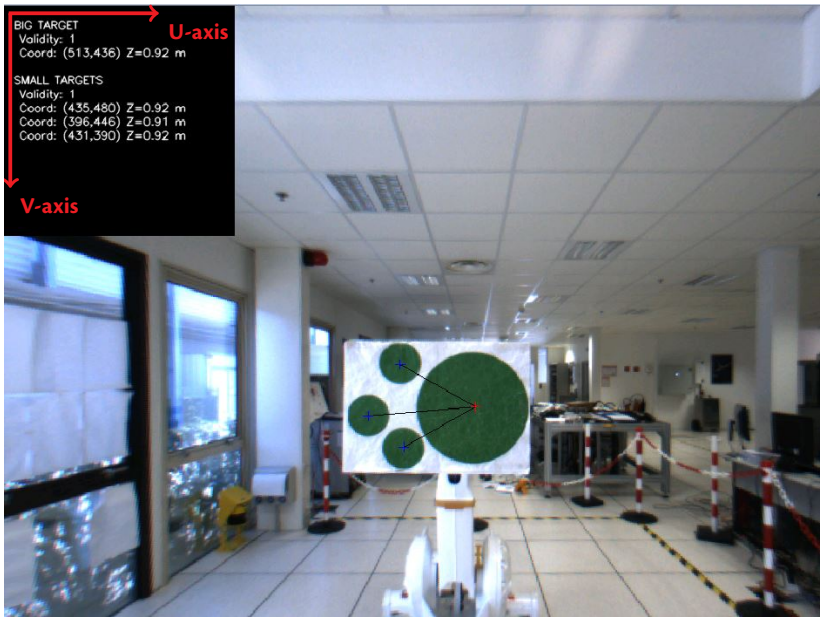


Figure 9.12: Image processing to identify the target position and attitude from the embedded camera

lyzed through the stereoscopic vision, and its distance from the camera is computed. These relative positions allow to deduce the relative attitude of the sheet of paper w.r.t. the camera.

A last detail must be mentioned concerning the relative position used in the feedback. The raw data coming from the camera and expressed in pixels are sent directly into a TCP/IP socket for the feedback. In order to use it in the control law, a conversion must be performed to obtain the vector expressed in meters. Denoting the planar position of the object in pixels by $(u, v)_{\text{pix}}$, and its distance from the camera expressed in meters by z, the planar position in meters is obtained by the following relations based on the classic Thales' theorem:

$$x = \frac{\|x\|\,\rho}{f}\left(u_{\text{pix}} - u^c_{\text{pix}}\right) \qquad \text{and} \qquad y = \frac{\|x\|\,\rho}{f}\left(v_{\text{pix}} - v^c_{\text{pix}}\right) \tag{9.2}$$

where the lateral position in meters are denoted by x and y, $f = 2.5\,\text{mm}$ is the lens focal distance, and $\rho = 4.65\,\mu\text{m}$ is the physical size of a pixel on the sensor. Besides, the frame tool_{C} of the camera is assumed to be located at the center of the two lenses and with the Z-axis along the line of sight. It means that the reference coordinates in pixels are given by $(u^c, v^c)_{\text{pix}}$ at the center of the image.

These considerations allow to write the transformation from the pixel space to the cartesian

frame tool_C as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \|z\|/f & 0 & 0 \\ 0 & \|z\|/f & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} u_{\text{pix}} \\ v_{\text{pix}} \\ z \end{bmatrix} - \begin{bmatrix} u^c_{\text{pix}} \\ v^c_{\text{pix}} \\ 0 \end{bmatrix} \right) \tag{9.3}$$

Each component of the test bench has been covered in the previous sections, from the network structure, to the industrial controllers, passing by the real-time simulator and the image processing used to closed the loop. The experimental results are presented in the following for a given test case.

9.2.3 Experimental results

The experimental results are presented at two levels. First, the delay introduced by the whole network communication has been investigated, in order to lower the expected controller performances. Then, the output of the experiments are given by the relative distance measured by the camera between the chaser's end-effector and the target point. The test case used in simulation is first detailed to emphasize which hypotheses have been made in order to perform a stable closed-loop, and then the experimental accuracy of the capture is presented.

9.2.3.1 Modeling hypotheses

The embedded simulation performs the capture of a small space debris by a chaser satellite, equipped with a robotic arm of 3DoF. The chaser data are taken from (Umetani and Yoshida, 1989b), with a base of 2 tons with a square basis of 3.5 meters, and a manipulator with two main segments of 2.5 meters each. The debris consists of a small cylinder-shaped spacecraft with a 75 cm radius, a 50 kg mass, and an initial angular rate of $11°/s$[1]. The goal of the simulation is to synchronize the chaser's effector with a target point located on the border of the debris. To do this, the "real world" dynamics which drives the robots motion is based on the previous simulator, as seen in 9.13. It is based on the following main hypothesis:

- Dynamics in weightlessness (orbital mechanics neglected);
- Rigid bodies only (no flexible panels or flexible segments);
- Perfect inertial sensors;
- Real camera feedback with noise and delay ;
- Perfect actuators;
- An optimal capture trajectory is computed a priori to the maneuver.

[1]This tumbling rate is among the fastest that are expected on-orbit, according to (Bonnal et al., 2013).

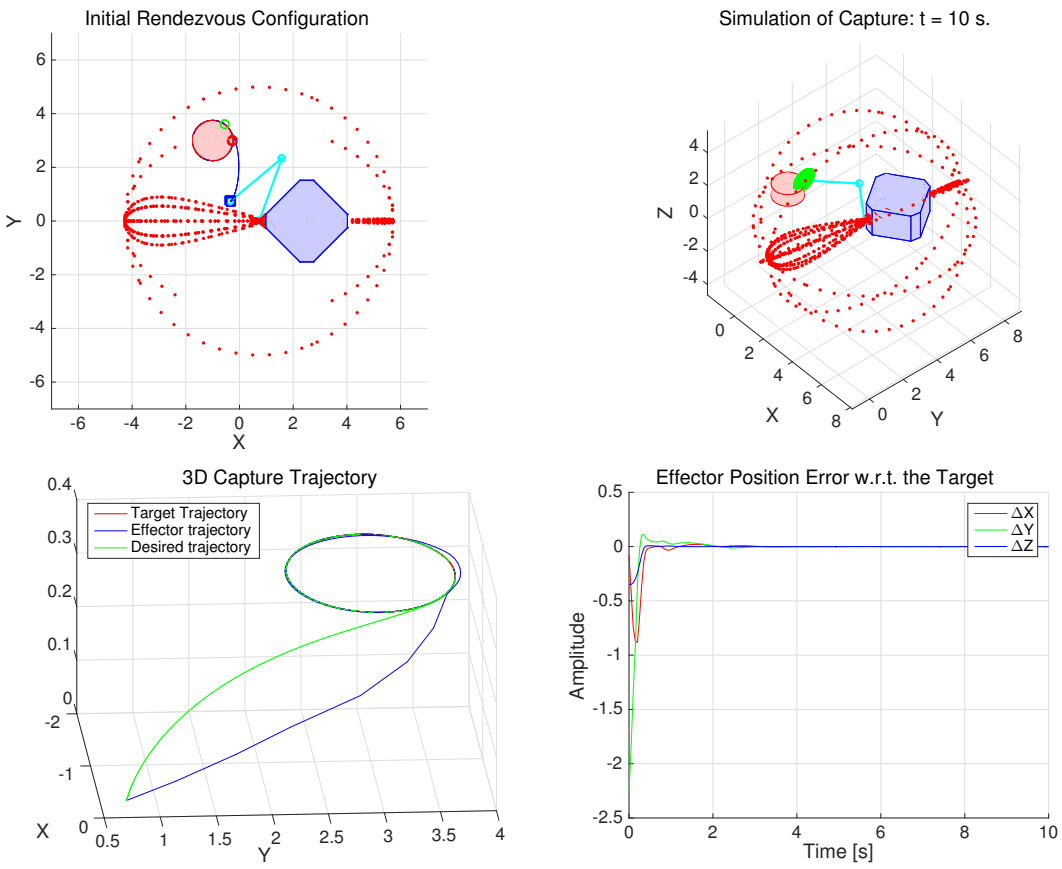


Figure 9.13: Visual interface of the simulator to check the on-orbit capture

This simplified model is used because the main goal was to validate the closed-loop simulation capability of the test bench. Later works in the company would benefit from all the work done to set up this test bench.

9.2.3.2 Results of the experiment

Before presenting the results, we briefly come back to the delay introduced by the network communication. This point is crucial for an efficient HIL since it determines the complexity of the dynamic scheme that can be used for the real-time simulation. The faster the communications are performed, the more complex the controller can be.

The main limitations on the closed-loop performances are coming from the communication delays. Very efficient control structures can be imagined and validated numerically, but once embedded on the dSpace module, the closed-loop is destabilized by the delay. In order to evaluate the delay of the industrial manipulators, commands were sent to them and received back, as illustrated in Figure 9.14. The analysis of these curves provide a mean delay of

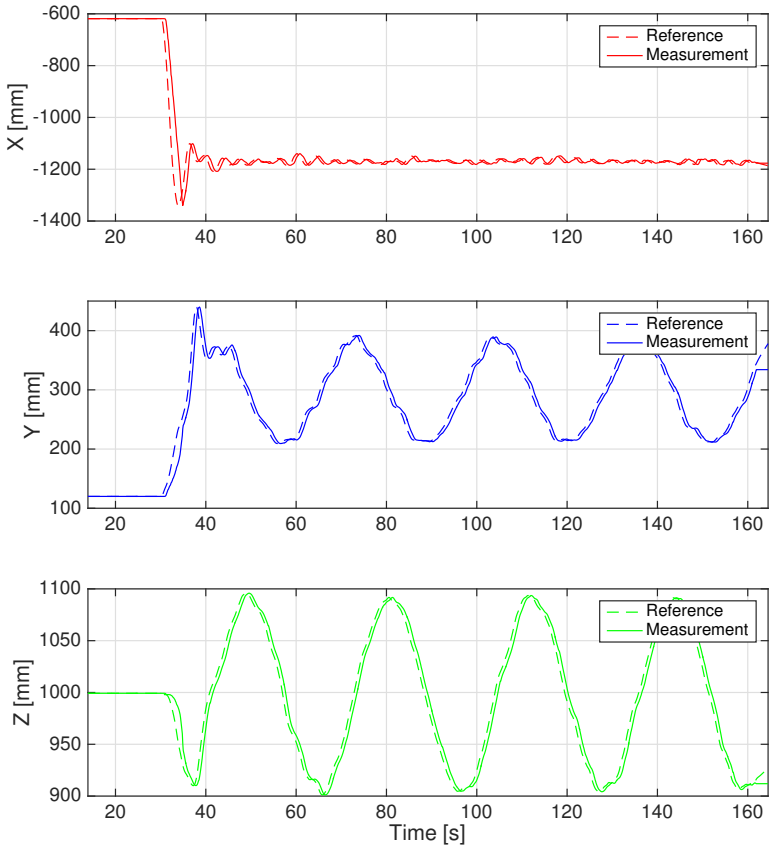


Figure 9.14: Delay in the IRC5 control loop of the industrial manipulator

approximately 80 ms. Therefore, the initial controller performance were reduced in terms of bandwidth, by reducing their gains. This approach was gradually performed by hand because it was almost impossible to predict in advance the additional delay required to compute the simulation itself at one time step. This latter was much depending on the closed loop complexity, and the instability may come from a simple additional filter in the loop.

Once the controller performance is set sufficiently low to prevent the system instability, the experiment of a capture is performed. The resulting accuracy for the example described above is presented in Figure 9.15. From the scheme in Figure 9.9, the world frame is represented in the (X, Z) plane, with the X-axis horizontal from the target to the chaser, the Z-axis upward, and the Y-axis to obtain a direct frame. Therefore, the lateral distances are given by the Y and Z signals in the plots, while the longitudinal one along the approach direction is given by the X signal. This latter is stabilized around the value of 2 m because a security offset was set between both manipulators to avoid any contact.

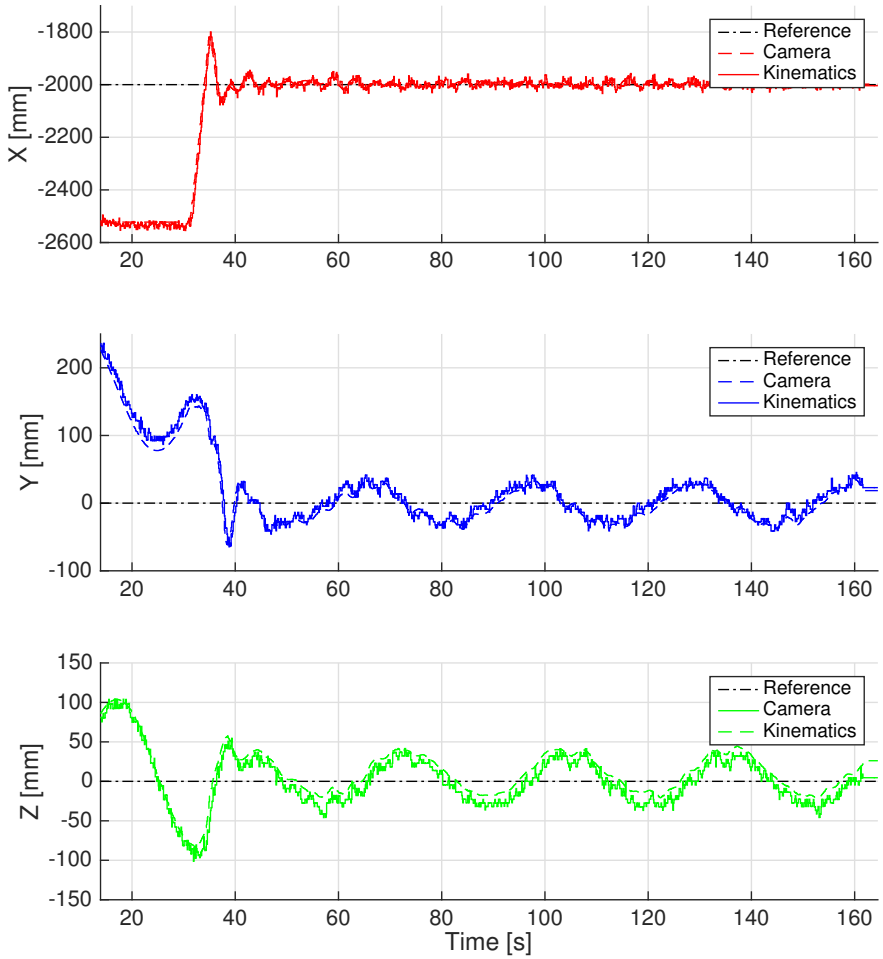


Figure 9.15: Vector components from the chaser's end-effector (i.e., the camera) to the target point (i.e., the biggest green circle)

From these results, the capture is validated with a mean error on the norm of the relative distance of 33.4 mm on a time horizon starting at 40 s, when the maneuver of capture seems completed. The resulting mean values, the errors and variances are summarized in Table 9.2. It can be seen that the mean value is actually very close to zero with a maximum bias of only 2 mm on the X-axis, while the mean error on this same axis rises to 1.1 cm. Similar mean values are obtained on the lateral axes, but a greater mean error. It means that the image processing is more accurate on the lateral axes than on the depth direction, but that it is more sensitive to the noise on these same lateral axes than the in-depth one. This conclusion is further strengthened by the variance results showing greater variations on the lateral axes. One explication to this phenomenon may be found in the transformation from pixels to distances in (9.3). Indeed, it is recalled that the lateral offsets in pixels are divided by the depth distance to obtain their values in meters. Therefore, a reasonable error on the in-depth value is magnified on the lateral ones by this operation.

Table 9.2: Mean error and variance of the relative distance during the capture

Axis	Value		Error	
	Mean	Variance	Mean	Variance
X	2.0 mm	210	11.3 mm	87
Y	0.1 mm	495	19.5 mm	116
Z	- 1.0 mm	505	19.7 mm	117

A second experiment was performed with the same simulation but at the calibration level. Carrying on the kinematic computations based on the measurements of the industrial robots, it is possible to assess the validity of the calibration data by comparing them with the image processing output. The results are plotted in Figure 9.15. It is clearly seen than the signal obtained by kinematic computations is very close to the camera one. The mean error between both signal on the norm of the global vector is only of 18.3 mm. Clearly, it seems quite intricate to decouple this error between the noise on the industrial robot measurements and the one coming from the camera. After all, the calibration of the test bench remains a key element to reduce as much as possible the sources of error. For comparison purpose, the EPOS test bench is able to perform a linear positioning of 0.3 mm and an angular one of 0.2°. With such positioning accuracy, the previous results would be directly linked to the camera noise and not to the industrial manipulators.

Conclusion and Improvements

As a conclusion, these experiments allowed to prove only partially all the dynamics and the control laws developed throughout the thesis, but their interest lies in the *Hardware-In-the-Loop (HIL)* capability with the potential validation of hardware components such as cameras or lasers. One main source of concern when performing such experiments are the computation time and the calibration data. The first one is critical to update at each time step both the on-orbit dynamics and the controller output based on a delayed and noisy measurement. The second point must be performed prior to any experiment, and using very accurate sensors inertially fixed in the room. Since the material was not at disposal at the time of this 6-months stay at TAS, an ad hoc calibration has been performed but may be one of the sources of errors.

As potential improvements for the future, many ideas can be found in the EPOS test bench (Boge and Ma, 2011). The current TAS test bench could be fully insulated to reproduce a darker background, closer to the space environment. A spot could be introduced to simulate the sun light coming from only one direction. More realistic mock-ups could be used to represent the debris on the target robot, in order to test more advanced image processing. A grapple mechanism could be mounted on the chaser as well to test the actual capture of an object. Doing this, a lot of care must be given to the mechanical closed loop arising at the instant of capture, since the robot may be damaged by too important efforts at the interface if a force control is not implemented.

CHAPTER 10 CONCLUSION AND RECOMMENDATIONS

10.1 Summary of contributions

This thesis went through the modeling and the control of a space robot with the aim of capturing a tumbling debris in orbit. The main contributions are summarized hereafter for the different topics involved in this project. Their advantages and limitations are emphasized before giving more advice for future works.

Modeling of flexible multi-body systems with a moving base The modeling of a multi-body system was investigated first for fixed-base structures, like industrial manipulators, in order to provide a common framework to describe their kinematics and dynamics with both rigid and flexible elements. A simplification of the flexible dynamics was proposed to reduce the computational burden of the simulations, while maintaining a high level of accuracy. Then, the dynamic model was augmented with the motion of the base undergoing external efforts. In the scope of space robots, the gravitational forces and the external on-orbit disturbances applying on this base were accounted for in the modeling phase. Using the *Decoupled Natural Orthogonal Complement (DeNOC)* framework, these steps were performed by using a local approach describing the dynamics of each element separately, in order to use the same model recursively for chain-like manipulators. As illustrated throughout the thesis, the models were written in a similar fashion that allowed to switch easily from the rigid to the flexible dynamics of an element, by simply updating its mass and twist-propagation matrices.

To perform the capture of a tumbling debris, the space robot must propagate the dynamics of the target point over time. To that end, a simple rigid model in rotation was proposed, knowing that a flexible one could be easily developed with the modeling tools introduced for the chaser.

Path planning with acceleration continuity at capture Assuming that the debris dynamics was accurately estimated, the space robot path planning consisted in finding the best way to move the end-effector toward the target point. A revisited path planner was introduced to provide the acceleration continuity at the instant of capture. The resulting trajectory aimed at performing a smoother capture, by allowing the end-effector to track the natural motion of the target before closing the grasping mechanism. Indeed, this requirement

on the acceleration yielded a continuous profile of the control efforts, meaning that a more robust capture was expected if uncertainties appear on the target dynamics.

This trajectory planning was based on the optimal control theory and was solved using the Pontryagin's minimum principle. The debris dynamics was considered explicitly, while the trajectory was supposed to be generated by its third order time derivative.

LFT modeling of the robotic mass matrix for robust stability and performance In order to follow this desired capture trajectory, a space robot controller was designed and synthesized in the thesis. A two level control law was proposed, with an inner loop allowing to efficiently reach a desired system configuration, while a second outer loop performed the global guidance. Two strategies for the inner loop were investigated with centralized and decentralized controllers, whose synthesis was based on the structured H_∞ framework. This approach allowed to specify a list of requirements in the frequency domain and to optimize the gains of a controller with a fixed structure. This was in sharp contrast with the older H_∞ methods that provide a full order controller. Lastly, the outer loop was built on a basic guidance scheme that coordinated the motion of the spacecraft with the motion of the manipulator, using the Jacobian matrix.

A strong contribution of the thesis was represented by the LFT modeling of any robotic system. A general formulation of the uncertain model of a rotation matrix w.r.t. its angle was first obtained under its minimal form. Then, this model was used recursively to describe any multi-body system w.r.t. the mechanical configuration of its joints. This tool was introduced for the first time to perform stability and performance μ-analyses on robotic manipulators. It allowed to derive the reduced ranges of motion of the joint angles, that ensured the performances for any arm configuration among them.

Closed-loop simulation on a robotic test bench with HIL Finally, the proposed control law for the space robot was partially validated on a robotic test bench with industrial manipulators. Numerical simulations were first performed to assess the validity of the control strategy, and to evaluate its computational burden as well. The delay required for its computation proved to be a critical feature for real-time simulation with hardware components. Bringing together the dynamic models and the control and guidance laws, the on-orbit behavior of the space robot and of the debris were obtained by numerical simulations, and reproduced on the robotic test bench. The final contribution of the thesis was the performance of a debris capture in weightlessness using a closed-loop scheme to guide the space robot with *Hardware-In-the-Loop (HIL)* components, namely a camera and its image processing software.

The main contributions of the thesis have been summarized above, and give rise to many more questions for future theoretical works and experiments. Some of them are proposed in the sequel.

10.2 Recommendations for future works

The current algorithms and control laws developed during this project may be improved manyfold to provide more efficient and more accurate strategies for the control of space robots. They are listed below according to the main topics of the thesis.

Modeling of tree-type systems and mechanical closed loop As mentioned in the modeling chapter, strong hypotheses were made on the space robot structure, with the assumption of a "star-shaped" system. Many improvements at the modeling level may be brought by extending the dynamic algorithms to tree-type systems according to the work of (Shah et al., 2012a). Doing this, the model of sophisticated structures could be obtained, like the *Dextre* manipulator on the ISS for example. Besides, closed-loop dynamics might be useful for the post-capture phase, and can still be obtained with the DeNOC framework (Saha et al., 2013). Indeed, the grasping of a debris by a multi-manipulator robot would lead to strong efforts at the interface. Though the modeling of such system in open-loop configuration is already possible with the algorithms in this thesis, their extension for the closed-loop configuration would be necessary for the study of the post-capture with a multi-arm chaser.

Another wide area of improvements is available for the modeling of the flexible behavior. Indeed, in-joint elasticities were not included in the thesis but may be considered as intermediate segments undergoing torsion. The impact of the presented distributed flexibility compared to this in-joint elasticity is worth studying. This analysis must provide more accurate models to design the controllers, and also to simulate the system in real-time by reducing dynamics to the main terms. In addition to the flexible modeling features, the algorithms may be extended for different representations of the flexibility, like the widespread *Finite Element Method (FEM)* instead of the AMM used throughout this thesis.

Lastly, dedicated elements for space system modeling could be developed from the works of this thesis. Indeed, additional models could be provided for CMGs and RWs by simplifying their dynamics using their rotational symmetry. The modeling of actuators and sensors was not introduced in this thesis and would contribute greatly to a better control of the real

systems. A graphical user interface could also make the link with a control engineer to model more easily any multi-body system with mixed rigid and flexible elements.

Path-planner with collision avoidance and improved positioning accuracy At the path-planning level, many works were mentioned in the literature review and could form the basis of future researches. Among those, the most critical topics that were not considered in this thesis were the collision avoidance and the actuators saturations. A good start is available in (Lampariello, 2013), where a single shooting method allows to solve the global optimization problem. The main focus of this research is the computation time since space systems always experience high delays due to the communication with the ground. An autonomous path planner may be an asset if it could be embedded on-board and computed in real-time by the space processor.

An interesting strategy for the planning could be to mix both the free-flying control for the approach phase, with the free-floating one during the capture phase, in order to avoid the disturbances on the manipulator from the base. Doing this, the planning must be adapted for each strategy and its continuity at the instant of switching is an interesting problem.

Improved synthesis schemes and robust analysis tools Concerning the control laws, the synthesis scheme used in the thesis was rather simple and could be augmented with additional performance transfers for the noise rejection and for the reference tracking. Similarly, a more complex controller could be developed by adding dynamics in its structure, for better trajectory tracking performances and to handle more efficiently the lowest flexible modes. The same synthesis method with the structured H_∞ framework may still be used.

At the robustness level, the LFT tools proved to be limited on the space robot system due to the high repetition of real uncertainties. Improved results may be expected by taking into account additional complex uncertainties, like delays or actuator dynamics. The main idea is to obtain worst cases by the computation of the lower bound in the μ-analysis. These latter would improve the controller robustness by performing a new multi-model synthesis that includes these worst configurations. For future works, there is also room for improvement to go further into the control and robustness analysis of the flexible systems.

More realistic test bench simulations Finally, the experimental validation of space robot control law is already an active field of research. Among the limitations on the work presented in the thesis, the delays experienced by the network communication was probably the most important one. In order to reduce as much as possible the impact of the hardware

components during the experiment, one first needs to evaluate precisely the delays introduced by each element; namely, the computation of the multi-body dynamics, the communication through the network, the time response of the industrial robots, etc. The quantification of the global delay expected in the feedback loop is crucial to reproduce faithfully the on-orbit kinematics and dynamics to test hardware components.

A last improvement to reach high-fidelity simulations is the realistic reproduction of the space environment, with a dark background and the use of a lighting spot to simulate the illumination conditions. These last features have tremendous effects on the image processing and, as a consequence, on the path planning based on this visual feedback.

BIBLIOGRAPHY

S. Abiko and K. Yoshida, "Post Flight Analysis of ETS-VII Space Robotic Experiments," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, vol. 1, no. 1, 2001, pp. 1–8.

F. Aghili, "Coordination Control of a Free-Flying Manipulator and Its Base Attitude to Capture and Detumble a Noncooperative Satellite," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, no. 1, 2009, pp. 2365–2372.

——, "Optimal Control of a Space Manipulator for Detumbling of a Target Satellite," in *IEEE International Conference on Robotics and Automation*, vol. 1, no. 1, 2009, pp. 3019–3024.

——, "A prediction and motion-planning scheme for visually guided robotic capturing of free-floating tumbling objects with uncertain dynamics," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 634 – 649, 2012. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6123215{&}tag=1

F. Aghili, M. Kuryllo, G. Okouneva, and C. English, "Fault-tolerant Pose Estimation of Space Objects," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, vol. 1. Ieee, jul 2010, pp. 947 – 954. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5695796

S. K. Agrawal, K. Pathak, J. Franch, R. Lampariello, and G. Hirzinger, "A Differentially Flat Open-Chain Space Robot with Arbitrarily Oriented Joint Axes and Two Momentum Wheels at the Base," *IEEE Transactions on Automatic Control*, vol. 54, no. 9, pp. 2185–2191, 2009.

B. A. Aikenhead, R. G. Daniell, and F. M. Davis, "Canadarm and the space shuttle," *Journal of Vacuum Science & Technology A*, vol. 1, no. 2, pp. 126 – 132, 1983. [Online]. Available: http://avspublications.org/jvsta/resource/1/jvtad6/v1/i2/p126{_}s1

D. Alazard, *Reverse engineering in control design.* London / Hoboken: ISTE / John Wiley & Sons, 2013.

D. Alazard and J.-P. Chrétien, "The impact of local masses and inertias on the dynamic modelling of flexible manipulators," in *Intelligence Artificielle Robotique et Automatique*

appliquée à l'Espace, vol. 1. Toulouse, France: CNES, 1992, pp. 1–15. [Online]. Available: http://oatao.univ-toulouse.fr/1956/

——, "Flexible joint control: Robustness analysis of the collocated and non-collocated feedbacks," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 3, 1993, pp. 2102–2107.

D. Alazard, C. Cumer, P. Apkarian, M. Gauvrit, and G. Ferreres, *Robustesse et commande optimale*. Toulouse: Cépadues-Editions, 1999.

D. Alazard, C. Cumer, and K. H. M. Tantawi, "Linear dynamic modeling of spacecraft with various flexible appendages and on-board angular momentums," in *7th International ESA Conference on Guidance, Navigation & Control Systems*, vol. 1, Tralee, Ireland, 2008, pp. 1–14. [Online]. Available: http://oatao.univ-toulouse.fr/1899/

D. Alazard, T. Loquen, H. de Plinval, and C. Cumer, "Avionics/Control co-design for large flexible space structures," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, vol. 1, 2013, pp. 1–15. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/6.2013-4638

F. Alby, J. Arnould, and A. Debus, *La pollution spatiale sous surveillance*. Ellipses, 2007.

H. Alexander and R. H. Cannon, "Experiments on the control of a satellite manipulator," *Progess in Material Handlings and Logistics*, vol. 1, no. 1, pp. 93–105, 1989.

M. Andrenucci, P. Pergola, and A. Ruggiero, "Active Removal of Space Debris - Expanding foam application for active debris removal," University of Pisa (Italy) - Aerospace Engineering Dept., Tech. Rep., 2011.

J. Angeles and S. K. Lee, "The Formulation of Dynamical Equations of Holonomic Mechanical Systems Using a Natural Orthogonal Complement," *Journal of Applied Mechanics (ASME)*, vol. 55, no. 1, pp. 243—-244, 1988. [Online]. Available: http://appliedmechanics.asmedigitalcollection.asme.org/article.aspx?articleid=1409200

F. Ankersen, "Guidance, Navigation, Control and Relative Dynamics for Spacecraft Proximity Maneuvers," Ph.D. dissertation, Aalborg University, 2011. [Online]. Available: http://vbn.aau.dk/ws/files/61599334/finnAnkersenThesis.pdf

M. Ansdell, "Active Space Debris Removal: Needs, Implications, and Recommendations for Today's Geopolitical Environment," *Journal of Public and International Affairs*, vol. 21, no. 1, p. 16, 2010.

P. Apkarian, "Tuning Controllers Against Multiple Design Requirements," in *American Control Conference*, vol. 1, 2013, pp. 3888–3893.

P. Apkarian and R. J. Adams, "Advanced Gain-Scheduling Techniques for Uncertain Systems," *IEEE Transactions on Control Systems Technology*, vol. 6, no. 1, pp. 21–32, 1998. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=654874

P. Apkarian and D. Noll, "Nonsmooth H_∞ synthesis," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 71–86, 2006.

M. Athans and P. L. Falb, *Optimal Control*. McGraw-Hill, 1966.

M. Benosman, F. Boyer, G. Le Vey, and D. Primault, "Flexible Links Manipulators : from Modelling to Control," *Journal of Intelligent and Robotic Systems*, vol. 34, no. 4, pp. 381–414, 2002.

T. Boge and O. Ma, "Using Advanced Industrial Robotics for Spacecraft Rendezvous and Docking Simulation," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=5980583

T. Boge, T. Wimmer, O. Ma, and T. Tzschichholz, "EPOS - Using Robotics for RvD Simulation of On-Orbit Servicing Missions," in *AIAA Modeling and Simulation Technologies Conference*, vol. 1, Toronto, Canada, 2010, pp. 1–15. [Online]. Available: http://elib.dlr.de/74383/

C. Bonnal, J.-M. Ruault, and M.-C. Desjean, "Active debris removal: Recent progress and current trends," *Acta Astronautica*, vol. 85, no. 1, pp. 51–60, 2013. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0094576512004602

W. J. Book, "Recursive Lagrangian Dynamics of Flexible Manipulator Arms," *The International Journal of Robotics Research*, vol. 3, no. 3, pp. 87–101, 1984. [Online]. Available: http://ijr.sagepub.com/content/3/3/87.short{#}cited-by

G. Boyarko, O. Yakimenko, and M. Romano, "Optimal Rendezvous Trajectories of a Controlled Spacecraft and a Tumbling Object," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 1239–1252, 2011. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/1.47645

F. Boyer and P. Coiffet, "Generalization of Newton-Euler model for flexible manipulators," *Journal of Robotic Systems*, vol. 13, no. 1, pp. 11–24, 1996.

F. Boyer and N. Glandais, "Simulation of flexible manipulators with elastic non-linearities," in *IEEE International Conference on Robotics and Automation*, vol. 1, Detroit, 1999, pp. 759–766. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=770066

F. Boyer, W. Khalil, M. Benosman, and G. Le Vey, "Modeling and Control of Flexible Manipulators," in *Modeling, Performance Analysis and Control of Robot Manipulators*. ISTE / John Wiley & Sons, 2010, vol. 1, no. 1, ch. 7, pp. 337—-394. [Online]. Available: http://www.control.isy.liu.se/research/reports/Ph.D.Thesis/PhD1349.pdf

S. Briot and W. Khalil, "Recursive Symbolic Calculation of the Dynamic Model of Flexible Parallel Robots," in *IEEE International Conference on Robotics and Automation*, vol. 1, 2013, pp. 5433–5438.

A. E. Bryson and Y. C. Ho, *Applied Optimal Control*. Hemisphere: New-York, 1975.

F. Caccavale and B. Siciliano, "Kinematic control of redundant free-floating robotic systems," *Advanced Robotics*, vol. 15, no. 4, pp. 429–448, jan 2001. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1163/156855301750398347

C. R. Carignan and D. L. Akin, "The Reaction Stabilization of On-Orbit Robots," *IEEE Control Systems*, vol. 20, no. 6, pp. 19 – 33, 2000. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=887446

M. D. Carpenter and M. A. Peck, "Reducing base reactions with gyroscopic actuation of space-robotic systems," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1262–1270, 2009.

J. Carusone, K. S. Buchan, and G. M. T. D'Eleuterio, "Experiments in End-Effector Tracking Control for Structurally Flexible Space Manipulators," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 553–560, 1993.

M. M. Castronuovo, "Active space debris removal - A preliminary mission analysis and design," *Acta Astronautica*, vol. 69, no. 9-10, pp. 848–859, nov 2011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0094576511001287

V. A. Chobotov, *Spacecraft Attitude Dynamics and Control*. Malaba, FL: Krieger, 1991.

J.-P. Chrétien, M. Delpech, and A. Louhmadi, "Modelling and simulation of distributed flexibillity in a spaceborne manipulator.pdf," in *IFAC Symposium on Automatic Control in Space*, vol. 1, 1985, pp. 269–307.

S. Cocuzza, S. Rossi, and S. Debei, "Robust reaction control of space manipulators," in *62nd International Astronautical Congress (IAC)*, vol. 1, 2011, pp. 627–639.

COPUOS, "Technical Report on Space Debris," Committee on the Peaceful Uses of Outer Space - United Nations Office for Outer Space Affairs, Tech. Rep., 1999.

——, "Space Debris Mitigation Guidelines," Committee on the Peaceful Uses of Outer Space - United Nations Office for Outer Space Affairs, Tech. Rep., 2010.

P. Couzin, R. Rembala, F. Teti, C. Bakouche, and C. Billot, "Active Removal of Large Debris : System Approach of Desorbiting Concepts and Technological Issues," in *6th European Conference on Space Debris*, vol. 2013, no. April, 2013, pp. 22–25.

J. J. Craig, *Introduction to Robotics : Mechanics and Control*, 2nd ed. Addison-Wesley Publishing Company, 1989.

X. Cyril, "Dynamics of Flexible-Link Manipulators," Ph.D. dissertation, McGill University, 1988.

C. Damaren and I. Sharf, "Simulation of Flexible-Link Manipulators With Inertial and Geometric Nonlinearities," *Journal of Dynamic Systems, Measurement, and Control*, vol. 117, no. 1, pp. 74–87, 1995. [Online]. Available: http://dx.doi.org/10.1115/1.2798525

A. De Luca and W. J. Book, *Robots with flexible elements*, springer v ed. Springer Handbook of Robotics, 2008.

A. De Luca and B. Siciliano, "Closed-Form Dynamic Model of Planar Multilink Lightweight Robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 4, pp. 826–839, 1991. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=108300

A. De Luca, P. Lucibello, and F. Nicolo, "Automatic symbolic modelling and nonlinear control of robots with flexible links," in *IEEE International Workshop on Robot Control: Theory and Applications*, vol. 1, 1988, pp. 62–70.

J. Denavit and R. S. Hartenberg, "A kinematic notation for lower pair mechanism based on matrices," *Journal of Applied Mechanics (ASME)1*, vol. 22, no. june, pp. 215–221, 1955.

M. A. Diftler, J. S. Mehling, M. E. Abdallah, N. A. Radford, L. B. Bridgwater, A. M. Sanders, R. S. Askew, D. M. Linn, J. D. Yamokoski, F. A. Permenter, B. K. Hargrave, R. Platt, R. T. Savely, and R. O. Ambrose, "Robonaut 2 - The first humanoid robot in

space," *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 2178–2183, 2011.

D. Dimitrov, "Dynamics and Control of Space Manipulators During a Satellite Capturing Operation," Ph.D. dissertation, Tohoku University, 2005.

S. Dubowsky and E. Papadopoulos, "The Kinematics, Dynamics, and Control of Free-Flying and Free-Floating Space Robotic Systems," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 531–543, 1993. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=258046

S. Dubowsky and M. A. Torres, "Path planning for space manipulators to minimize spacecraft attitude disturbances," in *IEEE International Conference on Robotics and Automation*, vol. 3, 1991, pp. 2522–2528. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=132005

S. Dubowsky, E. E. Vance, and M. A. Torres, "The Control of Space Manipulators Subject to Spacecraft Attitude Control Saturation Limits," in *NASA Conference on Space Telerobotics*, vol. IV, 1989, pp. 409–418.

S. K. Dwivedy and P. Eberhard, "Dynamic analysis of flexible manipulators, a literature review," *Mechanism and Machine Theory*, vol. 41, no. 7, pp. 749–777, 2006.

A. Ebrahimi and S. A. A. Moosavian, "Dynamics of Space Free-Flying Robots with Flexible Appendages," *Journal of Aerospace Science and Technology (JAST)*, vol. 4, no. 4, pp. 29–36, 2007. [Online]. Available: http://www.sid.ir/en/ViewPaper.asp?ID=150194{&}vDate=FALL2007{&}vEnd=36{&}vJournal=JOURNAL+OF+AEROSPACE+SCIENCE+AND+TECHNOLOGY+(JAST){&}vNo=4{&}vStart=29{&}vVolume=4{&}vWriter=EBRAHIMIA.,MOUSAVIANS.A.A.

B. Escudier and J.-Y. Pouillard, *Mécanique Spatiale*. Toulouse: École Nationale Supérieure de l'Aéronautique et de l'Espace (Supaero), 1996.

E. Fabacher, S. Lizy-Destrez, D. Alazard, and F. Ankersen, "Guidance and navigation for electromagnetic formation flight orbit modification," in *Proceedings of EuroGNC 2015*, Toulouse, FR, April 2015, pp. pp.1–20.

R. Featherstone, *Rigid Body Dynamics Algorithms*. Boston: Springer US, 2008. [Online]. Available: http://www.springerlink.com/index/10.1007/978-0-387-74315-8

R. Featherstone and D. E. Orin, "Robot Dynamics: Equations and Algorithms," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, San Francisco, 2000, pp. 826 – 834. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=844153

W. Fehse, *Automated RendezVous and Docking of Spacecraft*, C. U. Press, Ed. Cambridge Aerospace Series, 2008. [Online]. Available: https://keyholepublishing.com/projectprove/Grs/Spcrft/AutomatedRendezvous{&}Docking.pdf

A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, vol. 68, no. 1, pp. 1–26, 2014.

R. B. Friend, "Orbital Express Program Summary and Mission Overview," in *Sensors and Systems for Space Applications II*, vol. 6958. Howard, Richard T. Motaghedi, Pejmun, 2008, pp. 1–11. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=835879

B. Fritsche, H. Klinkrad, A. Kashkovsky, and E. Grinberg, "Spacecraft Disintegration During Uncontrolled Atmospheric Re-Entry," *Acta Astronauticaa*, vol. 47, no. 2-9, pp. 513–522, 2000.

M. Fuchs, C. Borst, P. Robuffo Giordano, A. Baumann, E. Kraemer, J. Langwald, R. Gruber, N. Seitz, G. Plank, K. Kunze, R. Burger, F. Schmidt, T. Wimboeck, and G. Hirzinger, "Rollin' Justin - Design considerations and realization of a mobile platform for a humanoid upper body," *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 4131–4137, 2009.

P. Gahinet and P. Apkarian, "Decentralized and fixed-structure H_∞ control in MATLAB," in *IEEE Conference on Decision and Control and European Control Conference*, vol. 1, 2011, pp. 8205–8210. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6160298

——, "Structured H_∞ Synthesis in MATLAB," in *IFAC World Congress*, vol. 18, 2011, pp. 1435–1440.

E. Garcia, M. A. Jimenez, P. Gonzalez De Santos, and M. Armada, "The Evolution of Robotics Research," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 90–103, 2007.

P. Gasbarri and A. Pisculli, "Dynamic/control interactions between flexible orbiting space-robot during grasping , docking and post-docking manoeuvres," *Acta Astronautica*, vol. 110, no. 1, pp. 225–238, 2015.

A. Grewal and V. Modit, "Dynamics and control of flexible multibody systems:an application to orbiting platforms," *IEEE International Conference on Systems,Man and Cybernetics*, vol. 3, no. 1, pp. 2093–2098, 1995.

C. Guariniello, L. Ansalone, and F. Curti, "Autonomous Capture of Non-Cooperative Spacecraft with a Space Free-Flyer," in *3rd CEAS Air and Space Conference*, vol. 1, 2011, pp. 1–10.

N. Guy, D. Alazard, C. Cumer, and C. Charbonnel, "Dynamic Modeling and Analysis of Spacecraft With Variable Tilt of Flexible Appendages," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 2, pp. 1–10, 2014.

S. M. Hashemi, H. S. Abbas, and H. Werner, "Low-complexity linear parameter-varying modeling and control of a robotic manipulator," *Control Engineering Practice*, vol. 20, no. 3, pp. 248–257, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.conengprac.2011.11.002

K. Hashtrudi Zaad and K. Khorasani, "Control of non-minimum phase singularly perturbed systems with application to flexible-link manipulators," *International Journal of Control*, vol. 63, no. 4, pp. 679–701, 1996. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/00207179608921863{#}preview

M. Hiltz, C. Rice, K. Boyle, and R. Allison, "Canadarm: 20 years of mission success through adaptation," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, vol. 1, 2001, pp. 1–8.

G. Hirzinger, K. Landzettel, B. Brunner, M. Fischer, C. Preusche, D. Reintsema, A. Albu-Schäffer, G. Schreiber, and B.-M. Steinmetz, "DLR's robotics technologies for on-orbit servicing," *Advanced Robotics*, vol. 18, no. 2, pp. 139–174, jan 2004. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1163/156855304322758006

J. M. Hollerbach, "A Recursive Lagrangian Formulation of Maniputator Dynamics and a Comparative Study of Dynamics Formulation Complexity," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-10, no. 11, pp. 730–736, 1980.

Y.-R. Hu and G. Vukovich, "Modeling and control of free flying flexible joint coordinated robots," in *International Conference on Advanced Robotics*, vol. 1, no. 1, 1997, pp. 1013–1020.

P. C. Hughes, *Spacecraft Attitude Dynamics*. New York: Wiley, 1986.

N. Inaba and M. Oda, "Autonomous Satellite Capture by a Space Robot : world first on-orbit experiment on a Japanese Robot Satellite ETS-VII," in *IEEE International Conference on Robotics & Automation*, vol. 2, 2000, pp. 1169 – 1174. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true{&}arnumber=844757{&}contentType=Conference+Publications

N. Inaba, M. Oda, and M. Hayashi, "Visual Servoing of Space Robot for Autonomous Satellite Capture," *Transactions of the Japan Society for Aeronautical and Space Sciences*, vol. 46, no. 153, pp. 173–179, 2003.

N. L. Johnson, "Statement before the Subcommittee on Space and Aeronautics at US House of Representatives," U.S. House of Representatives - http://gop.science.house.gov/Media/hearings/space09/april28/johnson.pdf, Tech. Rep., 2009. [Online]. Available: http://gop.science.house.gov/Media/hearings/space09/april28/johnson.pdf

T. Kasai, M. Oda, and T. Suzuki, "Results of the ETS-7 mission - Rendezvous, docking and space robotics experiments," in *5th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, vol. 1. Nordwijk, The Netherlands: ESTEC/ESA, 1999, pp. 299–306.

S. Kawamura, F. Miyazaki, and S. Arimoto, "Is a local linear PD feedback control law effective for trajectory tracking of robot motion ?" in *IEEE International Conference on Robotics and Automation*, vol. 3, 1988, pp. 1335–1340.

D. J. Kessler and B. G. Cour-Palais, "Collision Frequency of Artificial Satellites : The Creation of a Debris Belt," *Journal of Geophysical Research*, vol. 83, no. A6, pp. 2637–2646, 1978.

H. Khalil, *Nonlinear Systems*, 3rd ed. Pearson, 2001.

W. Khalil, "Dynamic modeling of robots using recursive Newton-Euler techniques," in *7th International Conference on Informatics in Control, Automation and Robotics*, vol. 1, 2010, pp. 19–31.

W. Khalil and J. F. Kleinfinger, "A new geometric notation for open and closed-loop robots," in *IEEE International Conference on Robotics and Automation*, vol. 3, 1986, pp. 1174–1179.

O. Khatib, "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

P. K. Khosla and T. Kanade, "Real-Time Implementation and Evaluation of Computed-Torque Scheme," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 2, pp. 245–253, 1989.

R. Koningstein and R. H. Cannon, "Experiments with Simplified Computed-Torque Controllers for Free-Flying Robots," in *American Control Conference*, vol. 1, 1991, pp. 1478–1484.

R. W. Krauss, "An Improved Technique for Modeling and Control of Flexible Structures," Ph.D. dissertation, Georgia Institute of Technology, 2006.

D.-S. Kwon and W. J. Book, "A Time-Domain Inverse Dynamic Tracking Control of a Single-Link Flexible Manipulator," *Journal of Dynamic Systems, Measurement, and Control*, vol. 116, no. June, p. 193, 1994.

R. Lampariello, "Motion Planning for the On-orbit Grasping of a Non-cooperative Target Satellite with Collision Avoidance," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, vol. 1, 2010, pp. 636–643.

——, "On Grasping a Tumbling Debris Object with a Free-Flying Robot," in *IFAC Symposium on Automatic Control in Aerospace*, vol. 19, no. 1, 2013, pp. 161–166. [Online]. Available: http://www.ifac-papersonline.net/Detailed/63171.html

D. J. Leith and W. E. Leithead, "Survey of gain-scheduling analysis and design," *International Journal of Control*, vol. 73, pp. 1001–1025, 2000.

J.-C. Liou, "An active debris removal parametric study for LEO environment remediation," *Advances in Space Research*, vol. 47, no. 11, pp. 1865–1876, jun 2011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0273117711000974

J.-C. Liou and N. L. Johnson, "A sensitivity study of the effectiveness of active debris removal in LEO," *Acta Astronautica*, vol. 64, no. 2, pp. 236–243, jan 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0094576508002634

J.-C. Liou, N. L. Johnson, and N. M. Hill, "Controlling the growth of future LEO debris populations with active debris removal," *Acta Astronautica*, vol. 66, no. 5-6,

pp. 648–653, mar 2010. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0094576509003981

M. Löhning, "LPV and LFR Modelling of Elastic Robots for Controller Synthesis," in *IEEE International Conference on Control and Automation*, vol. 1, 2010, pp. 522–527.

R. W. Longman, R. E. Lindberg, and M. F. Zedd, "Satellite-mounted robot manipulators - New kinematics and reaction moment compensation," *International Journal of Robotics Research*, vol. 6, no. 3, pp. 87–103, 1987.

S. Lopez-Linares, A. Konno, and M. Uchiyama, "Vibration Controllability of 3D Flexible Manipulators," *Journal of Dynamic Systems, Measurement and Control*, vol. 119, no. 2, pp. 326–330, 1997.

T. Loquen, H. de Plinval, C. Cumer, and D. Alazard, "Attitude control of satellites with flexible appendages : a structured H_∞ control design," in *AIAA Guidance, Navigation and Control Conference*, vol. 1, 2012, p. 18. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/6.2012-4845

M. Loudini, D. Boukhetala, and M. Tadjine, "Mathematical modelling of a single link flexible manipulator," in *International Control Conference*, vol. FA3, 2006, pp. 1–6.

Z. Ma, O. Ma, and B. N. Shashikanth, "Optimal approach to and alignment with a rotating rigid body for capture," *Journal of the Astronautical Sciences*, vol. 55, no. 4, pp. 407–419, 2007.

T. Macaluso, C. Bourdin, F. Buloup, M.-L. Mille, P. Sainton, F. R. Sarlegna, V. Taillebot, J.-L. Vercher, P. Weiss, and L. Bringoux, "Kinematic features of whole-body reaching movements underwater: Neutral buoyancy effects." *Neuroscience*, vol. 327, no. 7, pp. 125–135, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306452216300690

C. Manceaux-Cumer and J.-P. Chrétien, "Minimal LFT form of a spacecraft built up from two bodies," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, vol. 1, 2001, pp. 1–9. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/6.2001-4350

E. Martin, E. Dupuis, J.-C. Piedboeuf, and M. Doyon, "The TECSAS mission from a canadian perspective," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, vol. 1, 2005, pp. 1–8.

R. Masoudi and M. Mahzoon, "Maneuvering and Vibrations Control of a Free-Floating Space Robot with Flexible Arms," *Journal of Dynamic Systems, Measurement, and Control*, vol. 133, no. 5, pp. 1–8, 2011.

Y. Masutani, F. Miyazaki, and S. Arimoto, "Sensory feedback control for space manipulators," in *IEEE International Conference on Robotics and Automation*, vol. 3, 1989, pp. 1346–1351.

J. Mayo, J. Dominguez, and A. A. Shabana, "Geometrically Nonlinear Formulations of Beams in Flexible Multibody Dynamics," *Journal of Vibration and Acoustics*, vol. 117, no. 4, pp. 501–509, 1995. [Online]. Available: http://dx.doi.org/10.1115/1.2874490

L. Meirovitch, "State Equations of Motion for Flexible Bodies in Terms of Quasi-Coordinates," in *IFAC Symposium on Dynamics of Controlled Mechanical Systems*, vol. 1, 1989, pp. 37 – 48.

——, "Hybrid State Equations of Motion for Flexible Bodies in Terms of Quasi-Coordinates," *Journal of Guidance, Control and Dynamics*, vol. 14, no. 5, pp. 1008–1013, 1991.

——, *Fundamentals of Vibrations*, M. E. Series, Ed. McGraw-Hill International Edition, 2001.

C. Menon, S. Busolo, S. Cocuzza, A. Aboudan, A. Bulgarelli, C. Bettanini, M. Marchesi, and F. Angrilli, "Issues and solutions for testing free-flying robots," *Acta Astronautica*, vol. 60, no. 12, pp. 957–965, jun 2007. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0094576506004279

M. Merino, E. Ahedo, C. Bombardelli, H. Urrutxua, and J. Peláez, "Ion Beam Shepherd Satellite for Space Debris Removal," in *4th European Conference for Aerospace Sciences*, vol. 1, 2011, pp. 1–8.

A. Mohan and S. K. Saha, "A recursive, numerically stable, and efficient simulation algorithm for serial robots," *Multibody System Dynamics*, vol. 17, no. 4, pp. 291–319, feb 2007. [Online]. Available: http://link.springer.com/10.1007/s11044-007-9044-8

——, "A recursive, numerically stable, and efficient simulation algorithm for serial robots with flexible links," *Multibody System Dynamics*, vol. 21, no. 1, pp. 1–35, 2009.

S. A. A. Moosavian and E. Papadopoulos, "Control of Space Free-Flyers Using the Modified Transpose Jacobian Algorithm," in *IEEE International Conference on Intelligent*

Robot and Systems (IROS), vol. 3. Ieee, 1997, pp. 1500–1505. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=656557

——, "Explicit dynamics of space free-flyers with multiple manipulators via SPACEMAPLE," *Advanced Robotics*, vol. 18, no. 2, pp. 223–244, 2004.

——, "Free-flying robots in space: An overview of dynamics modeling, planning and control," *Robotica*, vol. 25, no. 5, pp. 537–547, mar 2007. [Online]. Available: http://www.journals.cambridge.org/abstract{_}S0263574707003438

——, "Modified transpose Jacobian control of robotic systems," *Automatica*, vol. 43, no. 7, pp. 1226–1233, jul 2007. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0005109807000878

H. H. S. Murali, D. Alazard, L. Massotti, F. Ankersen, and C. Toglia, *Advances in Aerospace Guidance, Navigation and Control: Selected Papers of the Third CEAS Specialist Conference on Guidance, Navigation and Control held in Toulouse.* Cham: Springer International Publishing, 2015, ch. Mechanical-Attitude Controller Co-design of Large Flexible Space Structures, pp. 659–678.

H. Nagamatsu, T. Kubota, and I. Nakatani, "Capture Strategy for Retrieval of a Tumbling Satellite by a Space Robotic Manipulator," in *IEEE International Conference on Robotics and Automation*, vol. 1, 1996, pp. 70–75.

T. Nagashio, T. Kida, T. Ohtani, and Y. Hamada, "Design and Flight Test Results of Robust Attitude Controller for the ETS-VIII Spacecraft," in *18th IFAC World Congress*, vol. 18, no. 1, 2011, pp. 5154–5159. [Online]. Available: http://www.ifac-papersonline.net/Detailed/49173.html

Y. Nakamura and R. Mukherjee, "Nonholonomic path planning of space robots," in *IEEE International Conference on Robotics and Automation*, vol. 2, 1989, pp. 1050–1055.

——, "Nonholonomic Path Planning of Space Robots via a Bidirectional Approach," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 4, pp. 500–514, 1991.

——, "Exploiting Nonholonomic Redundancy of Free-Flying Space Robots," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 4, pp. 499–506, 1993.

K. Nanos and E. Papadopoulos, "On Cartesian Motions with Singularities Avoidance for Free-floating Space Robots," in *IEEE International Conference on Robotics and*

Automation, vol. 1, 2012, pp. 5398–5403. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6224695

——, "Avoiding dynamic singularities in Cartesian motions of free-floating manipulators," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 2305–2318, 2015.

NASA, "On-orbit satellite servicing study," Satellite Servicing Capabilities Office - NASA, Tech. Rep., 2010. [Online]. Available: http://ssco.gsfc.nasa.gov/images/NASA{_}SatelliteServicing{_}Project{_}Report{_}0511.pdf

D. N. Nenchev, "Reaction Null Space of a multibody system with applications in robotics," *Mechanical Sciences*, vol. 4, no. 1, pp. 97–112, feb 2013. [Online]. Available: http://www.mech-sci.net/4/97/2013/

D. N. Nenchev, Y. Umetani, and K. Yoshida, "Analysis of a Redundant Free-Flying Spacecraft/Manipulator System," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pp. 1–6, 1992. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=127234

D. N. Nenchev, K. Yoshida, P. Vichitkulsawat, and M. Uchiyama, "Reaction Null-Space Control of Flexible Structure Mounted Manipulator Systems," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 6, pp. 1011–1023, 1999.

T.-C. Nguyen-Huynh and I. Sharf, "Adaptive Reactionless Motion for Space Manipulator when Capturing an Unknown Tumbling Target," in *IEEE International Conference on Robotics and Automation*, vol. 1, 2011, pp. 4202 – 4207. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=5980398

S. Nicosia, F. Nicolo, and D. Lentini, "Dynamical control of industrial robots with elastic and dissipative joints," in *8th IFAC World Congress*, vol. 1, 1981, pp. 1933–1939.

S. Nicosia, P. Valigi, and L. Zaccarian, "Dynamic modelling of a two link flexible robot and experimental validation," in *IEEE International Conference on Robotics and Automation*, vol. 3, no. April, 1996, pp. 1953–1958.

C. M. Oakley and R. H. Cannon, "End-Point Control of a Two-Link Manipulator with a Very Flexible Forearm: Issues and Experiments," in *American Control Conference*, 1989, pp. 1381–1389.

J. Obermark, G. Creamer, B. E. Kelm, W. Wagner, and C. G. Henshaw, "SUMO/FREND: Vision System for Autonomous Satellite Grapple," in *Sensors and Systems for Space Applications*, R. T. Howard and R. D. Richards, Eds., vol. 6555, 2007, pp. 1–11. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1302383

M. Oda, "On the dynamics and control of ETS-7 satellite and its robot arm," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 1994, pp. 1586 – 1593. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=407644

——, "Coordinated control of spacecraft attitude and its manipulator," in *IEEE International Conference on Robotics and Automation*, vol. 1, 1996, pp. 732–738. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=503861

——, "Motion control of the satellite mounted robot arm which assures satellite attitude stability," *Acta Astronautica*, vol. 41, no. 11, pp. 739–750, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094576597002142{#}

——, "Coordinated control of a satellite mounted robot arm and the satellite attitude," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, vol. 1, 1999, pp. 626–632. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/6.1999-4062

T. Oki, H. Nakanishi, and K. Yoshida, "Time-Optimal Manipulator Control of a Free-Floating Space Robot with Constraint on Reaction Torque," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2008, pp. 2828–2833.

N. Ortiz Gómez, S. J. Walker, M. Jankovic, J. M. Romero Martín, F. Kirchner, and M. L. Vasile, "Control analysis for a contactless de-tumbling method based on eddy currents: problem definition and approximate proposed solutions," in *AIAA Guidance, Navigation, and Control Conference*, no. January, 2016, pp. 1–25. [Online]. Available: http://arc.aiaa.org/doi/10.2514/6.2016-0642

C. Ott, A. Albu-Schäffer, A. Kugi, and G. Hirzinger, "On the Passivity-Based Impedance Control of Flexible Joint Robots," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 416–429, 2008.

E. Papadopoulos and S. Dubowsky, "On the dynamic singularities in the control of free-floating space manipulators," in *ASME Winter Annual Meeting*, vol. 15, 1989, pp. 45–52.

——, "Coordinated manipulator/spacecraft motion control for space robotic systems," in *IEEE International Conference on Robotics and Automation*, vol. 2, 1991, pp.

1696–1701. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true{&}arnumber=131864{&}contentType=Conference+Publications

——, "On the nature of control algorithms for free-floating space manipulators," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 750–758, 1991. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=105384

E. Papadopoulos and S. A. A. Moosavian, "Dynamics and Control of Multi-arm Space Robots During Chase and Capture Operations," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 3, no. 1. Ieee, 1994, pp. 1554–1561. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=407648

E. Papadopoulos and I. Paraskevas, "Design and configuration control of space robots undergoing impact," in *International ESA Conference on Guidance, Navigation and Control Systems*, vol. 1, 2006, pp. 1–10. [Online]. Available: ftp://home.dei.polimi.it/outgoing/Marco.Lovera/ESAGNC05/papers/p14{_}papa.pdf

E. Papadopoulos, I. Poulakakis, and I. Papadimitriou, "On Path Planning and Obstacle Avoidance for Nonholonomic Platforms with Manipulators: A Polynomial Approach," *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 367–383, apr 2002. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/027836402320556377

C. Pardini, T. Hanada, and P. H. Krisko, "Benefits and risks of using electrodynamic tethers to de-orbit spacecraft," *Acta Astronautica*, vol. 64, no. 5-6, pp. 571–588, mar 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0094576508003433

K. Parsa, "The Lagrangian Derivation of Kane's Equations," *Transactions of the Canadian Society for Mechanical Engineering*, vol. 31, no. 4, pp. 407–420, 2007.

L. Pedersen, D. Kortenkamp, D. Wettergreen, and I. Nourbakhsh, "A Survey of Space Robotics," in *7th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, vol. 1, 2003, pp. 1–8.

S. M. Persson, "Autonomous and safe capture of large space debris with a robotic manipulator," Ph.D. dissertation, McGill University, 2015.

C. R. Phipps, "A laser-optical system to re-enter or lower low Earth orbit space debris," *Acta Astronautica*, vol. 93, pp. 418–429, jan 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0094576513002749

C. R. Phipps, G. Albrecht, H. Friedman, D. Gavel, E. V. George, J. Murray, C. Ho, W. Priedhorsky, M. M. Michaelis, and J. P. Reilly, "ORION : Clearing near-Earth space debris using a 20kW, 530nm, Earth-based, repetitively pulsed laser," *Laser and Particle Beams*, vol. 14, no. 1, pp. 1–44, 1996.

J.-C. Piedboeuf, "Introduction à la modélisation des robots flexibles," Tech. Rep., 1992.

P. Piersigilli, I. Sharf, and A. K. Misra, "Reactionless capture of a satellite by a two degree-of-freedom manipulator," *Acta Astronautica*, vol. 66, no. 1-2, pp. 183–192, jan 2010. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0094576509003361

P. Rank, Q. Mühlbauer, W. Naumann, and K. Landzettel, "The DEOS Automation and Robotics Payload," in *11th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, no. April, 2011, pp. 1–8. [Online]. Available: http://robotics.estec.esa.int/ASTRA/Astra2011/Papers/04B/FCXNL-11A06-2139219-1-2139219rank.pdf

D. Reintsema, J. Thaeter, A. Rathke, W. Naumann, P. Rank, and J. Sommer, "DEOS - The German Robotics Approach to Secure and De-Orbit Malfunctioned Satellites from Low Earth Orbits," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, vol. 1, 2010, pp. 244–251.

I. Rekleitis, E. Martin, G. Rouleau, R. L'Archevêque, K. Parsa, and E. Dupuis, "Autonomous capture of a tumbling satellite," *Journal of Fields Robotics*, vol. 24, no. 4, pp. 1–22, 2007.

G. Rodriguez, "Kalman Filtering , Smoothing , and Recursive Robot Arm Forward and Inverse Dynamics," *IEEE Journal of Robotics and Automation*, vol. 3, no. 6, pp. 624–639, 1987.

G. Rodriguez, A. Jain, and K. Kreutz-Delgado, "A Spatial Operator Algebra for Manipulator Modeling and Control," *The International Journal of Robotics Research*, vol. 10, no. 4, pp. 371–381, 1991.

C. Roos, "Systems modeling, analysis and control (SMAC) toolbox: An insight into the robustness analysis library," in *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*, 2013, pp. 176–181.

C. Roos, F. Lescher, J. M. Biannic, C. Döll, and G. Ferreres, "A set of μ-analysis based tools to evaluate the robustness properties of high-dimensional uncertain systems," in *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*, 2011, pp. 644–649.

B. Rubinger, M. Brousseau, J. Lymer, C. Gosselin, T. Laliberté, and J.-C. Piedboeuf, "A Novel Robotic Hand-SARAH for Operations on the International Space Station," in *7th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, vol. 1, no. 1, 2002, pp. 1–8.

W. J. Rugh and J. S. Shamma, "Research on gain-scheduling," *Automatica*, vol. 36, pp. 1401–1425, 2000.

T. Rybus and K. Seweryn, "Trajectory planning and simulations of the manipulator mounted on a free-floating satellite," in *Aerospace Robotics*. Springer Berlin Heidelberg, 2013, vol. 1, ch. Selected P, pp. 61–73. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84894632616{&}partnerID=tZOtx3y1

T. Rybus, J. Nicolau-Kukliński, K. Seweryn, T. Barciński, M. Ciesielska, K. Grassmann, J. Grygorczuk, M. Karczewski, M. Kowalski, M. Krzewski, T. Kuciński, J. Lisowski, R. Przybyla, K. Skup, T. Szewczyk, and R. Wawrzaszek, "New planar air-bearing microgravity simulator for verification of space robotics numerical simulations and control algorithms," in *12th ESA Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, vol. 1, 2013, pp. 1–8.

J.-H. Ryu, D.-S. Kwon, and B. Hannaford, "Control of a Flexible Manipulator with Noncollocated Feedback : Time Domain Passivity Approach," in *Spinger tracts in advanced robotics*. Springer-Verlag Berlin Heidelberg, 2003, pp. 121–134. [Online]. Available: http://link.springer.com/chapter/10.1007/3-540-36224-X{_}8

M. Saad, "Modélisation et passivité d'un système à un bras flexible," Ph.D. dissertation, Ecole Polytechnique de Montréal, 2003.

M. Sabatini, P. Gasbarri, R. Monti, and G. B. Palmerini, "Vibration control of a flexible space manipulator during on orbit operations," *Acta Astronautica*, vol. 73, no. 1, pp. 109–121, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.actaastro.2011.11.012

S. S. Sachdev, "Canadarm - a review of its flights," *Journal of Vacuum Science & Technology A*, vol. 4, no. 3, pp. 268 –272, may 1986. [Online]. Available: http://link.aip.org/link/?JVA/4/268/1{&}Agg=doi

S. K. Saha, "A Decomposition of the Manipulator Inertia Matrix," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 301–304, 1997.

——, "Dynamics of serial multibody systems using the decoupled natural orthogonal complement matrices," *Journal of applied mechanics*, vol. 66, no. 4, pp. 986–996, 1999.

[Online]. Available: http://appliedmechanics.asmedigitalcollection.asme.org/article.aspx?articleid=1413891

S. K. Saha, S. V. Shah, and P. V. Nandihal, "Evolution of the DeNOC-based dynamic modelling for multibody systems," *Mechanical Sciences*, vol. 4, no. 1, pp. 1–20, 2013. [Online]. Available: http://www.mech-sci.net/4/1/2013/

F. Saupe and H. Pfifer, "An Observer Based State Feedback LFT LPV Controller for an Industrial Manipulator," in *IFAC Symposium on Robust Control Design*, vol. 7, no. 1, 2012, pp. 337–342.

W. O. Schall, "Removal of small space debris with orbiting lasers," in *SPIE Conference on High-Power Laser Ablation*, vol. 3343, 1998, pp. 564–574.

P. B. Selding, "Space Station Required No Evasive Maneuvers in 2013 Despite Growing Debris Threat," 2014. [Online]. Available: http://spacenews.com/39121space-station-required-no-evasive-maneuvers-in-2013-despite-growing-debris/

K. Seweryn, M. Banaszkiewicz, B. Maediger, T. Rybus, and J. Sommer, "Dynamics of space robotic arm during interactions with non-cooperative objects," in *11th ESA Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, vol. 1. Noordwijk, The Netherlands: ESTEC, 2011, pp. 1–8.

A. A. Shabana, "Flexible Multibody Dynamics : Review of Past and Recent Developments," *Multibody System Dynamics*, vol. 1, no. 2, pp. 189–222, 1997. [Online]. Available: http://link.springer.com/article/10.1023/A:1009773505418

S. V. Shah, P. V. Nandihal, and S. K. Saha, "Recursive dynamics simulator (ReDySim): A multibody dynamics solver," *Theoretical and Applied Mechanics Letters*, vol. 2, no. 6, pp. 1—-6, 2012. [Online]. Available: http://link.aip.org/link/TAMLBX/v2/i6/p063011/s1{&}Agg=doi

S. V. Shah, S. K. Saha, and J. K. Dutt, "Modular framework for dynamic modeling and analyses of legged robots," *Mechanism and Machine Theory*, vol. 49, no. 1, pp. 234–255, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.mechmachtheory.2011.10.006

I. Sharf and C. Damaren, "Simulation of flexible-link manipulators: Basis functions and nonlinear terms in the motion equations," *IEEE International Conference on Robotics and Automation*, vol. 1, no. 1, pp. 1956–1962, 1992.

B. Siciliano and W. J. Book, "A Singular Perturbation Approach to Control of Lightweight Flexible Manipulators," *The International Journal of Robotics Research*, vol. 7, no. 4, pp. 79–90, 1988.

S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 2nd ed. John Wiley & Sons, 2005.

J. R. Spofford, "Redundant actuator coordination for a free-flying telerobot," in *American Control Conference*, vol. 1, 1991, pp. 1497–1502.

M. W. Spong, "Modeling and control of elastic joint robots," *Journal of Dynamic Systems, Measurement and Control (ASME)*, vol. 37, no. 11, pp. 1782–1786, 1992.

M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. New York: John Wiley & Sons, 2006. [Online]. Available: http://ca.wiley.com/WileyCDA/WileyTitle/productCd-EHEP000518.html

E. Staffetti, H. Bruyninckx, and J. De Schutter, *On the Invariance of Manipulability Indices*. Dordrecht: Springer Netherlands, 2002, pp. 57–66.

S. Stamm and P. Motaghedi, "Orbital Express Capture System: concept to reality," in *Proceedings of the SPIE Conference on Spacecraft Platforms and Infrastructure*, vol. 5419, 2004, pp. 78–91.

E. Stoll, S. Jaekel, J. Katz, A. Saenz-Otero, and R. Varatharajoo, "SPHERES Interact - Human-Machine Interaction aboard the International Space Station," *Journal of Fields Robotics*, vol. 29, no. 4, pp. 554–575, 2012.

B. Sullivan, B. Kelm, G. Roesler, and C. G. Henshaw, "DARPA Robotic Space Servicer: On-Demand Capabilities in GEO," in *AIAA SPACE Conference and Exposition*, 2015, pp. 1–24.

K. H. M. Tantawi, "Linear dynamic modeling of satellites with various flexible appendages," Master Thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace - SupAero, 2007.

K. K. Timmons and J. C. Ringelberg, "Approach and Capture for Autonomous Rendezvous and Docking," in *IEEE Aerospace Conference*, 2008, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4526515

S. P. Timoshenko, *Vibration problems in engineering*, 3rd ed. Van Nostrand, 1955.

M. A. Torres and S. Dubowsky, "Minimizing Spacecraft Attitude Distubrances in Space Manipulator Systems," *Journal of Guidance, Control and Dynamics*, vol. 15, no. 4, pp. 1010–1017, 1992.

——, "Path-planning for Elastically Constrained Space Manipulator Systems," in *IEEE International Conference on Robotics and Automation*, vol. 1, 1993, pp. 812 – 817. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=292077{&}tag=1

Y. Umetani and K. Yoshida, "Experimental study on two-dimensionnal free-flying robot satellite model," in *NASA Conference on Space Telerobotics*, vol. 5, 1989, pp. 215–224.

——, "Resolved Motion Rate Control of Space Robotic Manipulators with Generalized Jacobian Matrix," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 303–314, 1989. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=34766

——, "Workspace and Manipulability Analysis of Space Manipulator," *Transactions of the Society of Instrument and Control Engineers*, vol. E-1, no. 1, pp. 1–8, 2001.

P. B. Usoro, R. Nadira, and S. S. Mahil, "A Finite Element/Lagrange Approach to Modeling Lightweight Flexible Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 3, pp. 198–205, 1986.

Z. Vafa and S. Dubowsky, "On the dynamics of manipulators in space using the virtual manipulator approach," in *IEEE International Conference on Robotics and Automation*, vol. 4, 1987, pp. 579–585. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=1088032

M. Vakil, R. Fotouhi, and P. N. Nikiforuk, "A New Method for Dynamic Modeling of Flexible-Link Flexible-Joint Manipulators," *Journal of Vibration and Acoustics*, vol. 134, no. 1, pp. 1–11, 2012.

E. E. Vance and R. M. Sanner, "Adaptive coordinated control of space manipulators using neural networks," in *Proceedings of the 13th World Congress in Automatic Control (IFAC)*, vol. P - Aerosp, 1996, pp. 147–152.

J. Vince, "Chapter 7 - Quaternions," in *Matrix Transforms for Computer Games and Animation*. London: Springer London, 2012, ch. 7, pp. 117–145. [Online]. Available: http://www.springerlink.com/index/10.1007/978-1-4471-4321-5

M. W. Walker and D. E. Orin, "Efficient Dynamic Computer Simulation of Robotic Mechanisms," *Journal of Dynamic Systems, Measurement and Control*, vol. 104, no. 3, pp. 205 – 211, 1982. [Online]. Available: http://dynamicsystems.asmedigitalcollection.asme.org/article.aspx?articleid=1403331

M. W. Walker and L.-B. Wee, "An Adaptive Control Strategy for Space Based Robot Manipulator," in *IEEE International Conference on Robotics and Automation*, vol. 2, 1991, pp. 1673–1680. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=131860

F. Wang, F. Sun, and H. Liu, "Space Robot Modeling and Control Considering the Effect of Orbital Mechanics," in *1st International Symposium on Systems and Control in Aerospace and Astronautics*, vol. 1, 2006, pp. 193 – 198. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=1627609

B. Weeden, "Overview of the legal and policy challenges of orbital debris removal," *Space Policy*, vol. 27, no. 1, pp. 38–43, feb 2011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0265964610001268

J. R. Wertz, D. F. Everett, and J. J. Puschell, *Space Mission Engineering: The New SMAD*. Microcosm Press, 2011.

Wikipedia, "Véhicule automatique de tranfert européen," 2013. [Online]. Available: http://fr.wikipedia.org/wiki/Véhicule_automatique_de_transfert_européen

——, "H-II Transfer Vehicle," 2013. [Online]. Available: http://fr.wikipedia.org/w/index.php?title=H-II_Transfer_Vehicle&action=history

J. D. Woods, "Space Junk Forces Space Station Crew to Seek Shelter," 2015. [Online]. Available: http://time.com/3961500/international-space-station-debris/

K. Wormnes, R. Le Letty, L. Summerer, H. Krag, R. Schonenborg, O. Dubois-Matra, E. Luraschi, A. Cropp, H. Krag, and J. Delaval, "ESA technologies for space debris remediation," *6th European Conference on Space Debris*, vol. 1, pp. 1–8, 2013. [Online]. Available: http://goo.gl/x32d9K

D. Wright, "The Current Space Debris Situation," p. 31, 2010.

W. Xu, B. Liang, Y. Xu, C. Li, and W. Qiang, "A Ground Experiment System of Free-floating Robot For Capturing Space Target," *Journal of Intelligent*

and Robotic Systems, vol. 48, no. 2, pp. 187–208, dec 2007. [Online]. Available: http://link.springer.com/10.1007/s10846-006-9087-8

W. Xu, D. Meng, Y. Chen, H. Qian, and Y. Xu, "Dynamics modeling and analysis of a flexible-base space robot for capturing large flexible spacecraft," *Multibody System Dynamics*, vol. 32, no. 3, pp. 357–401, 2014.

Y. Xu, "The measure of dynamic coupling of space robot systems," in *IEEE International Conference on Robotics and Automation*, vol. 3, 1993, pp. 615–620. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=291837

Y. Xu, H.-Y. Shum, T. Kanade, and J.-J. Lee, "Parameterization and Adaptive Control of Space Robot Systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 2, pp. 435–451, 1994.

K. Yamada, T. Yoshikawa, and Y. Fujita, "Arm path planning of a space robot with angular momentum," *Journal of Advanced Robotics*, vol. 9, no. 6, pp. 693–709, 1995.

K. Yoshida, "Achievements in Space Robotics," *IEEE Robotics & Automation Magazine*, vol. 16, no. 4, pp. 20–28, dec 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5306922

K. Yoshida and H. Nakanishi, "Impedance Matching in Capturing a Satellite by a Space Robot," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 3-4, 2003, pp. 3059–3064. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1249626

K. Yoshida and Y. Umetani, "Control of space free-flying robot," in *29th Conference on Decision and Control*, vol. 1, 1990, pp. 97–102. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=203553

K. Yoshida and B. Wilcox, "Space Robots and Systems," in *Springer Handbook of Robotics*. Springer, 2008, ch. 45, pp. 1031–1063.

K. Yoshida, D. Dimitrov, and H. Nakanishi, "On the Capture of Tumbling Satellite by a Space Robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1. Ieee, oct 2006, pp. 4127–4132. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4059057

T. Yoshikawa and K. Hosoda, "Modeling of Flexible Manipulators Using Virtual Rigid Links and Passive Joints," in *IEEE International Workshop on Intelligent*

Robots and Systems, vol. 2. Ieee, 1991, pp. 967–972. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=174614

APPENDIX A RIGID MODELING DETAILS

This appendix gathers all the results and computations used in the modeling of rigid manipulators.

First, the *Denavit Hartenberg (DH)* parameters are described in appendix A.1, according to the two main convention available in the literature: the classic and the modified conventions. A new adapted convention is also developed in order to place the joint frames at a the segment base. This will be more suited to describe dynamics of elbow-shaped segments in flexible modeling.

Then, the traditional formula relating Euler angles and their corresponding angular rate or accelerations are derived in appendix A.2. The similar transformations are also presented for the quaternions. Even if they are only little covered in the modeling, they are often used in simulation so this reminder may appear to be useful though.

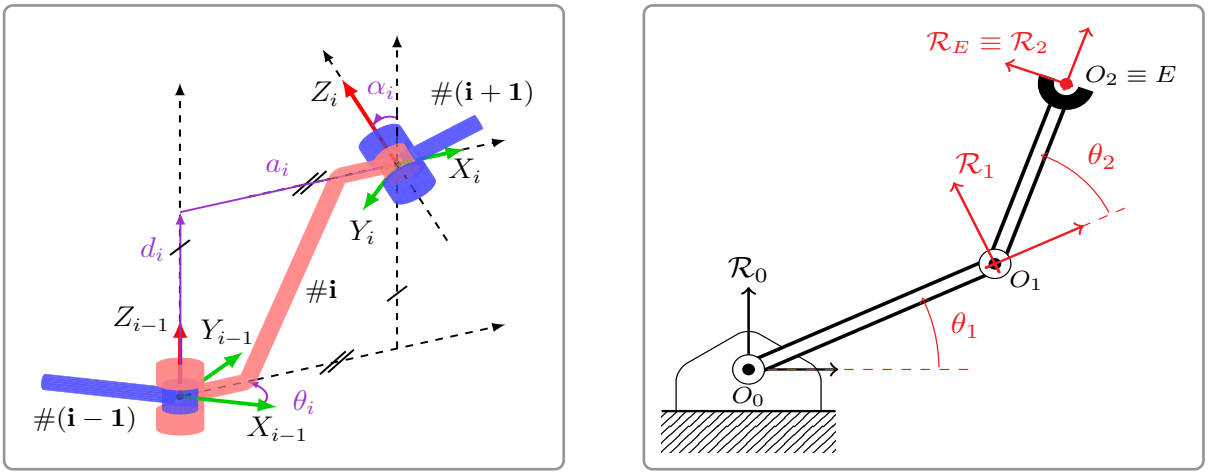
Finally, all the dynamics algorithms are presented in appendix A.3. It is of the utmost importance to notice that the demonstrations are performed for the general case using the standard matrices: ${}^{i}R_{i+1}$, $\mathbf{M}$, $\mathbf{A}_{i,i-1}$ and $\mathbf{P}_i$. Therefore, the resulting algorithms are valid for both rigid and flexible cases, provided that they are updated with the corresponding model.

A.1 Denavit-Hartenberg Parameters

A.1.1 Classic Denavit-Hartenberg parameters

Introduced in (Denavit and Hartenberg, 1955), the classic *Denavit Hartenberg (DH)* parameters are defined to compute any robot kinematics in a systematic way. They define a set of four parameters for each segment $\{\theta_i, d_i, a_i, \alpha_i\}$, which describes the translation and rotation from the frame $\mathcal{R}_{i-1}$ to the frame $\mathcal{R}_i$, rigidly fixed to the i^{th} segment. With the DH convention, the Z-axis of $\mathcal{R}_{i-1}$ is aligned with the i^{th} joint axis. It results that its origin O_i is most often located at the segment's end-tip, as illustrated for a planar example in Figure A.1b and for a manipulator with six joints in Figure A.4a. The i^{th} frame $\mathcal{R}_i = (O_i|X_i, Y_i, Z_i)$ is set as follows:

- Z_i is aligned with the $(i+1)^{th}$ joint axis;
- X_i is along the common perpendicular of Z_i and Z_{i-1};
- Y_i is chosen to have a direct frame;
- O_i is taken at the intersection of Z_i with the common perpendicular of Z_i and Z_{i-1}.



(a) Kinematic scheme (b) Planar example

Figure A.1: Classic Denavit-Hartenberg parameters

The kinematic scheme between two successive segments is described by the next steps (see Figure A.1a):

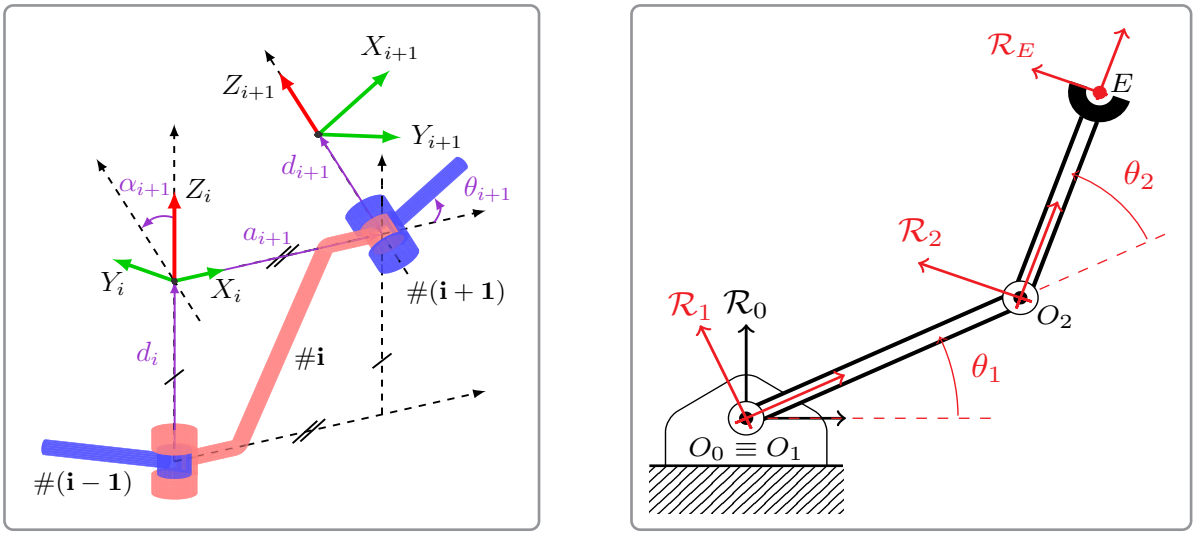
1. Rot(θ_i,Z): with θ_i the angle $\angle(X_{i-1}, X_i)$ about Z_{i-1};
2. Trans(d_i,Z): with d_i the distance $\langle O_{i-1}, X_i \rangle$ along Z_{i-1};
3. Trans(a_i,X): with a_i the distance $\langle Z_{i-1}, O_i \rangle$ along X_i;
4. Rot(α_i,X): with α_i the angle $\angle(Z_{i-1}, Z_i)$ about X_i.

A.1.2 Modified Denavit-Hartenberg parameters

Modified parameters were introduced by W. Khalil and J. F. Kleinfinger (Khalil and Kleinfinger, 1986) to overcome the ambiguous cases arising for tree-type systems. Indeed, the i^{th} frame is oriented according to the next segment axis in the classic DH convention, so it becomes unclear when there are multiple segments attached to it. Modified DH parameters are defined in a similar way but the Z-axis of $\mathcal{R}_i$ is aligned with the i^{th} joint axis, instead of the $(i-1)^{th}$ one. The difference for the planar example is illustrated in Figure A.2b, and for the 3D model in Figure A.4b.

With the Khalil convention, the frame $\mathcal{R}_i = (O_i|X_i, Y_i, Z_i)$ is set as follows:

- Z_i is aligned with the i^{th} joint axis;
- X_i is along the common perpendicular of Z_i and Z_{i+1};
- Y_i is chosen to have a direct frame;
- O_i is taken at the intersection of Z_i with the common perpendicular of Z_i and Z_{i+1}.



(a) Kinematic scheme (b) Planar example

Figure A.2: Modified Denavit-Hartenberg parameters

The modified DH parameters define the transformation between two following frames according to the kinematic scheme illustrated in Figure A.2a. Such a transformation is performed by following the next steps:

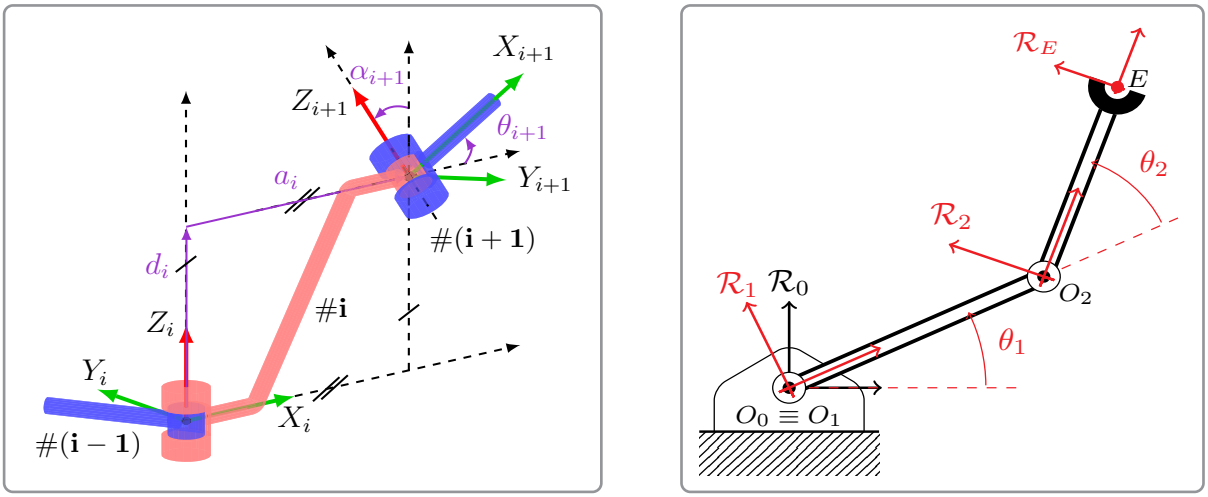
1. Rot(α_{i+1},X): with α_{i+1} the angle $\angle(Z_i, Z_{i+1})$ about X_i;
2. Trans(a_{i+1},X): with a_{i+1} the distance $\langle O_i, Z_{i+1}\rangle$ along X_i;
3. Trans(d_{i+1},Z): with d_{i+1} the distance $\langle X_i, O_{i+1}\rangle$ along Z_{i+1};
4. Rot(θ_{i+1},Z): with θ_{i+1} the angle $\angle(X_i, X_{i+1})$ about Z_{i+1}.

It must be noted the following equivalence with the classic parameters illustrated in Table A.1:

$$\left[\begin{array}{cccc} \alpha_{i+1} & a_{i+1} & d_{i+1} & \theta_{i+1} \end{array}\right]_{Modified} = \left[\begin{array}{cccc} \alpha_{i} & a_{i} & d_{i+1} & \theta_{i+1} \end{array}\right]_{Classic}$$

A.1.3 Adapted Denavit-Hartenberg parameters

A similar convention is used in this thesis, with a slight change to ensure that $\mathcal{R}_i$ is rigidly fixed at segment's base and not at its end-tip E or at an imaginary point like it happens for the Khalil convention. Derived from this latter, the new convention, called *adapted*, is more suited to describe the flexible segment considered in the thesis. **Indeed, when considering elbow-shaped segments the new convention makes frame $\mathcal{R}_i$ coinciding with the *floating frame of reference* $\mathcal{R}_{o_i}$.** For example, with Khalil convention, when the joint axis is along the segment's length and when d_i parameter is not zero, the frame $\mathcal{R}_i$ is not fixed



(a) Kinematic scheme (b) Planar example

Figure A.3: Adapted Denavit-Hartenberg parameters

at segment's base O but at its end-tip E, as illustrated for the first segment in Figure A.4b. With this new convention, frame $\mathcal{R}_i$ is set at the base, as shown in Figure A.4c.

This adapted convention is mainly based on the Khalil one. The same rules are used to choose the axes (X_i, Y_i, Z_i) but O_i is set differently :

- O_i is taken at the intersection of Z_i with the common perpendicular of Z_{i-1} and Z_i.

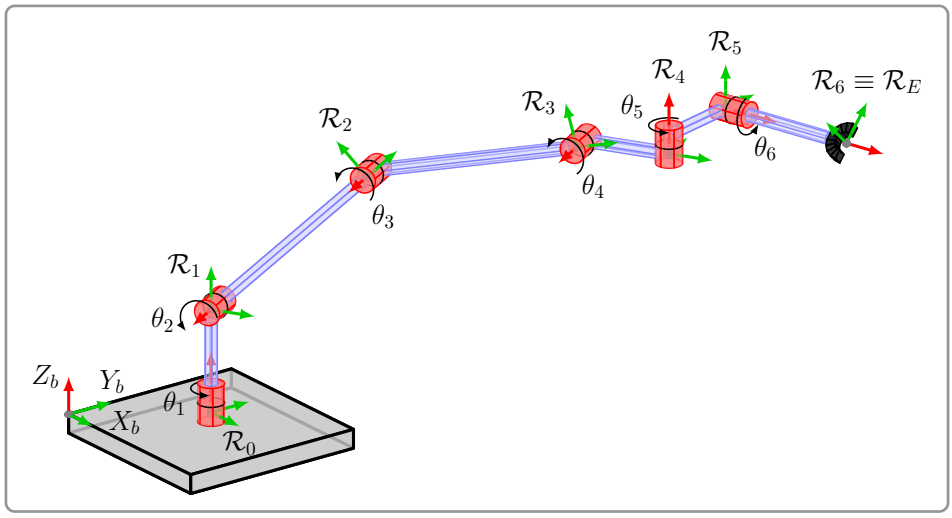
The Adapted DH parameters are defined for a new kinematic scheme given in Figure A.3a:

1. Trans(d_{i+1},Z): with d_{i+1} the distance $\langle O_i, Z_{i+1}\rangle$ along Z_i;
2. Trans(a_{i+1},X): with a_{i+1} the distance $\langle O_i, Z_{i+1}\rangle$ along X_i;
3. Rot(α_{i+1},X): with α_{i+1} the angle $\angle(Z_i, Z_{i+1})$ about X_i;
4. Rot(θ_{i+1},Z): with θ_{i+1} the angle $\angle(X_i, X_{i+1})$ about Z_{i+1}.

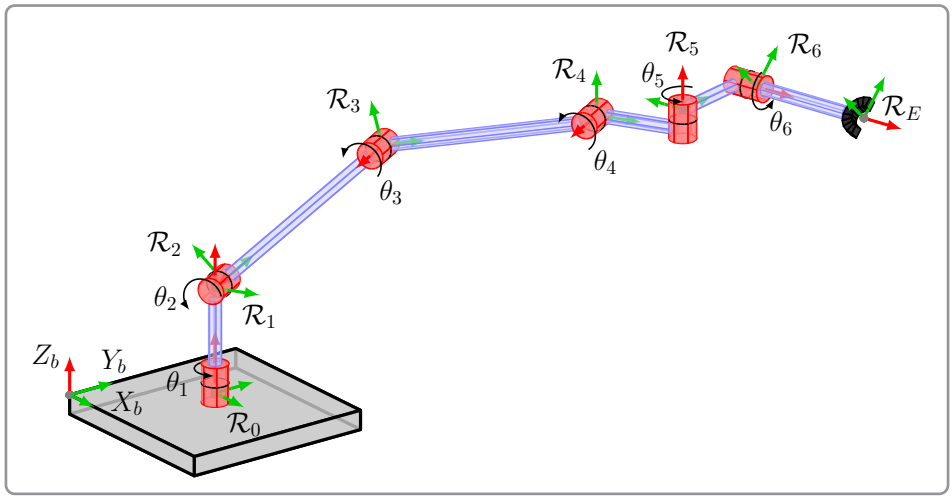
It must be noted the following equivalence with the classic parameters illustrated in Table A.1:

$$\left[\begin{array}{cccc} d_{i+1} & a_{i+1} & \alpha_{i+1} & \theta_{i+1} \end{array}\right]_{Adapted} = \left[\begin{array}{cccc} d_i & a_i & \alpha_i & \theta_{i+1} \end{array}\right]_{Classic}$$

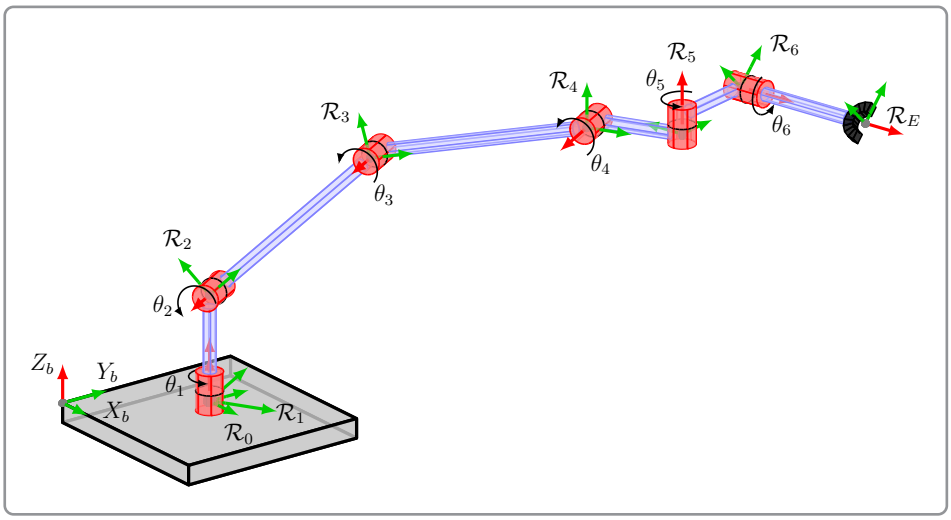
For example, the DH parameters are compared for each convention: the classic, the modified, and the adapted one. It yields the following results when the tool frame is described as the $(n+1)^{th}$ frame. Segment lengths are denoted by l_i, and joint lengths by d, which are supposed identical for each joint.



(a) Classic Denavit-Hartenberg convention



(b) Modified Denavit-Hartenberg convention



(c) Adapted Denavit-Hartenberg convention

Figure A.4: Different Denavit-Hartenberg frames depending on the convention

Table A.1: Denavit-Hartenberg parameters for the Canadarm in Figure A.4

Classical parameters					**Khalil parameters**					**Adapted parameters**				
#	$\boldsymbol{\theta}_\mathbf{i}$	$\mathbf{d_i}$	$\mathbf{a_i}$	$\boldsymbol{\alpha}_\mathbf{i}$	#	$\boldsymbol{\alpha}_\mathbf{i}$	$\mathbf{a_i}$	$\mathbf{d_i}$	$\boldsymbol{\theta}_\mathbf{i}$	#	$\mathbf{d_i}$	$\mathbf{a_i}$	$\boldsymbol{\alpha}_\mathbf{i}$	$\boldsymbol{\theta}_\mathbf{i}$
1	0	l_1	0	$\pi/2$	**1**	0	0	l_1	0	**1**	0	0	0	$-\pi/2$
2	0	$-d$	l_2	0	**2**	$\pi/2$	0	$-d$	0	**2**	l_1	0	$\pi/2$	0
3	0	$-d$	l_3	0	**3**	0	l_2	$-d$	0	**3**	$-d$	l_2	0	0
4	0	$-d$	l_4	$-\pi/2$	**4**	0	l_3	$-d$	0	**4**	$-d$	l_3	0	0
5	$\pi/2$	d	l_5	$\pi/2$	**5**	$-\pi/2$	l_4	d	$\pi/2$	**5**	$-d$	l_4	$-\pi/2$	$\pi/2$
6	π	l_6	0	0	**6**	$\pi/2$	l_5	0	π	**6**	d	l_5	$\pi/2$	π
E	0	0	0	0	**E**	0	0	l_6	0	**E**	l_6	0	0	0

A.2 Euler Angles and Quaternions

Euler angles and quaternions are two of the main representations of attitude to describe system dynamics (Hughes, 1986). Euler angles are also widely used in aeronautics because they have a direct physical meaning ro represent the angular variations around the nominal attitude of an aircraft. On the other hand, when angular motion are greater, singularities may occur with the Euler representations and a fourth parameter is introduce to avoid them. In that sense, quaternions provide a powerful tool for numerical simulation since they stay stable for all range of angular motion, and describe rotations by only four parameters.

In the following, rotation matrices are written for both representations. Then, the relation is given between their time derivative and the angular rate of the corresponding body, as well as the relation between their acceleration and the time derivative of the angular rate itself.

A.2.1 Direct Cosine Matrix

Let us denote the angular motion of a body called "b" by two representations, namely by the Euler angles with $\boldsymbol{\Psi}_\mathrm{b}$, and by the quaternions by $\mathcal{Q}_\mathrm{b}$. The rotation matrix expresses the transformation of a vector from one frame to another. It is denoted by ${}^{\mathrm{I}}R_\mathrm{b}$ to transform a vector ${}^{(\mathrm{b})}\mathbf{x}$ expressed in frame $\mathcal{R}_\mathrm{b}$ into ${}^{(\mathrm{I})}\mathbf{x}$ expressed in frame $\mathcal{R}_\mathrm{I}$. One of the main properties of rotation matrices is that their inverse is obtained by transposing them, since they belong to the special orthogonal group $\mathcal{SO}(3)$. The transformation of a vector ${}^{(\mathrm{I})}\mathbf{x}$ from $\mathcal{R}_\mathrm{I}$ into $\mathcal{R}_\mathrm{b}$ is thus given by ${}^{\mathrm{b}}R_\mathrm{I} = {}^{\mathrm{I}}R_\mathrm{b}^\top$. Therefore, only one rotation matrix is developed in the sequel, the one projecting a local vector $\mathbf{x}$ from the body-fixed frame to the inertial one:

$$ {}^{(\mathrm{I})}\mathbf{x} = {}^{\mathrm{I}}R_\mathrm{b}\, {}^{(\mathrm{b})}\mathbf{x} \tag{A.1} $$

For the quaternion, the Hamilton product is denoted "$\otimes$", so the definition of a change of frame reads as follows (Vince, 2012):

$$^{(\mathrm{I})}\mathbf{x} = \mathcal{Q}_{\mathrm{b}} \otimes {}^{(\mathrm{b})}\mathbf{x} \otimes \mathcal{Q}_{\mathrm{b}}^{-1} \tag{A.2}$$

Euler angles The rotation matrix corresponding to an Euler angle parameterization is given here for the "yaw-pitch-roll" order (or 3-2-1 Euler convention) by:

$$\begin{aligned} {}^{\mathrm{I}}R_{\mathrm{b}} &= \mathrm{Rot}(\Psi_z, Z)\,\mathrm{Rot}(\Psi_y, Y)\,\mathrm{Rot}(\Psi_x, X) \\ &= \begin{bmatrix} c_y\,c_z & -c_x\,s_z + s_x\,s_y\,c_z & s_x\,s_z + c_x\,s_y\,c_z \\ c_y\,s_z & c_x\,c_z + s_x\,s_y\,s_z & -s_x\,c_z + c_x\,s_y\,s_z \\ -s_y & s_x\,c_y & c_x\,c_y \end{bmatrix} \end{aligned} \tag{A.3}$$

where $c_i = \cos(\Psi_i)$ and $s_i = \sin(\Psi_i)$ for $i \in \{x; y; z\}$, and denoting the Euler angles as follows:

$$\mathbf{\Psi}_{\mathrm{b}} = \begin{bmatrix} \Psi_x & \Psi_y & \Psi_z \end{bmatrix}^\top \tag{A.4}$$

Quaternion The same rotation matrix based on quaternions is expressed by:

$${}^{\mathrm{I}}R_{\mathrm{b}} = \begin{bmatrix} 2(u^2 + v_x^2) - 1 & 2(v_x\,v_y - u\,v_z) & 2(v_x\,v_z + u\,v_y) \\ 2(v_x\,v_y + u\,v_z) & 2(u^2 + v_y^2) - 1 & 2(v_y\,v_z - u\,v_x) \\ 2(v_x\,v_z - u\,v_y) & 2(v_y\,v_z + u\,v_x) & 2(u^2 + v_z^2) - 1 \end{bmatrix} \tag{A.5}$$

where the quaternion scalar/vector parts are denoted, respectively, u and $\boldsymbol{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^\top$, as follows:

$$\mathcal{Q}_{\mathrm{b}} = \begin{bmatrix} u & v_x & v_y & v_z \end{bmatrix}^\top \tag{A.6}$$

A.2.2 Angular Rate

When describing the dynamics of a rigid body, the angular rate $\boldsymbol{\omega}_{\mathrm{b}}$ is often preferred as state variable compared to the time derivative of the corresponding Euler angles or quaternion. But these latter are necessary to represent the resulting attitude of the body. The angular rate is traditionally obtained by integrating the dynamic equations of motion, and then, used to compute the time derivative of the chosen angular representation before integrating it in time.

In order to build a full simulation scheme, one needs the transformation matrix between these

two quantities. Is it worth recalling that the following matrices are not symmetric anymore, and that they may not be invertible for some singular configurations.

Euler angles With the Euler angles representation, the simulation scheme is represented by the next steps (Craig, 1989; Spong et al., 2006):

$$\dot{\boldsymbol{\omega}}_{\mathrm{b}} \qquad \Longrightarrow \qquad \boldsymbol{\omega}_{\mathrm{b}} = R_{\boldsymbol{\Psi}_{\mathrm{b}}}\,\dot{\boldsymbol{\Psi}}_{\mathrm{b}} \qquad \Longrightarrow \qquad \boldsymbol{\Psi}_{\mathrm{b}} = \int_t R_{\boldsymbol{\Psi}_{\mathrm{b}}}^{-1}\,\boldsymbol{\omega}_{\mathrm{b}}\,\mathrm{d}t \tag{A.7}$$

The frame of expression of the angular rate must also be considered carefully because it entirely changes the transformation. For ${}^{(\mathrm{I})}\boldsymbol{\omega}_{\mathrm{b}}$ expressed in the inertial frame, it results in:

$${}^{(\mathrm{I})}R_{\boldsymbol{\Psi}_{\mathrm{b}}} = \begin{bmatrix} c_z\,c_y & -s_z & 0 \\ s_z\,c_y & c_z & 0 \\ -s_y & 0 & 1 \end{bmatrix} \qquad \text{and} \qquad {}^{(\mathrm{I})}R_{\boldsymbol{\Psi}_{\mathrm{b}}}^{-1} = \begin{bmatrix} c_z/c_y & s_z/c_y & 0 \\ -s_z & c_z & 0 \\ s_y\,c_z/c_y & s_y\,s_z/c_y & 1 \end{bmatrix} \tag{A.8}$$

On the other hand, if ${}^{(\mathrm{b})}\boldsymbol{\omega}_{\mathrm{b}}$ is expressed in the body frame:

$${}^{(\mathrm{b})}R_{\boldsymbol{\Psi}_{\mathrm{b}}} = \begin{bmatrix} 1 & 0 & -s_y \\ 0 & c_x & s_x\,c_y \\ 0 & -s_x & c_x\,c_y \end{bmatrix} \qquad \text{and} \qquad {}^{(\mathrm{b})}R_{\boldsymbol{\Psi}_{\mathrm{b}}}^{-1} = \begin{bmatrix} 1 & s_y\,s_x/c_y & s_y\,c_x/c_y \\ 0 & c_x & -s_x \\ 0 & s_x/c_y & c_x/c_y \end{bmatrix} \tag{A.9}$$

Quaternion When the quaternion representation is used, the relation with the angular rate expressed in the inertial frame reads as follows:

$$\dot{\boldsymbol{\omega}}_{\mathrm{b}} \qquad \Longrightarrow \qquad \dot{\mathcal{Q}}_{\mathrm{b}} = \frac{1}{2}\begin{bmatrix} 0 \\ {}^{(\mathrm{I})}\boldsymbol{\omega}_{\mathrm{b}} \end{bmatrix} \otimes \mathcal{Q}_{\mathrm{b}} \qquad \Longrightarrow \qquad \mathcal{Q}_{\mathrm{b}} = \int_t \frac{1}{2}\begin{bmatrix} 0 \\ {}^{(\mathrm{I})}\boldsymbol{\omega}_{\mathrm{b}} \end{bmatrix} \otimes \mathcal{Q}_{\mathrm{b}}\,\mathrm{d}t \tag{A.10}$$

leading to a set of differential equations for the quaternion terms:

$$\begin{aligned} \dot{u} &= \frac{1}{2}\left(-v_x\,{}^{(\mathrm{I})}\omega_x - v_y\,{}^{(\mathrm{I})}\omega_y - v_z\,{}^{(\mathrm{I})}\omega_z\right) \\ \dot{v}_x &= \frac{1}{2}\left(\ u\ {}^{(\mathrm{I})}\omega_x - v_y\,{}^{(\mathrm{I})}\omega_z + v_z\,{}^{(\mathrm{I})}\omega_y\right) \\ \dot{v}_y &= \frac{1}{2}\left(\ u\ {}^{(\mathrm{I})}\omega_y + v_x\,{}^{(\mathrm{I})}\omega_z - v_z\,{}^{(\mathrm{I})}\omega_x\right) \\ \dot{v}_z &= \frac{1}{2}\left(\ u\ {}^{(\mathrm{I})}\omega_z - v_x\,{}^{(\mathrm{I})}\omega_y + v_y\,{}^{(\mathrm{I})}\omega_x\right) \end{aligned}$$

On the other hand, if the angular rate is expressed in the body-fixed frame, the product is

done in the opposite way:

$$\dot{\boldsymbol{\omega}}_{\mathrm{b}} \quad \Longrightarrow \quad \dot{\mathcal{Q}}_{\mathrm{b}} = \frac{1}{2}\,\mathcal{Q}_{\mathrm{b}} \otimes \begin{bmatrix} 0 \\ {}^{(\mathrm{b})}\boldsymbol{\omega}_{\mathrm{b}} \end{bmatrix} \quad \Longrightarrow \quad \mathcal{Q}_{\mathrm{b}} = \int_t \frac{1}{2}\,\mathcal{Q}_{\mathrm{b}} \otimes \begin{bmatrix} 0 \\ {}^{(\mathrm{b})}\boldsymbol{\omega}_{\mathrm{b}} \end{bmatrix} \mathrm{d}t \quad \text{(A.11)}$$

and the differential equations are derived by:

$$\begin{aligned}
\dot{u} &= \frac{1}{2}\left(-v_x\,{}^{(\mathrm{b})}\omega_x - v_y\,{}^{(\mathrm{b})}\omega_y - v_z\,{}^{(\mathrm{b})}\omega_z\right) \\
\dot{v}_x &= \frac{1}{2}\left(\ \ u\ \ {}^{(\mathrm{b})}\omega_x + v_y\,{}^{(\mathrm{b})}\omega_z - v_z\,{}^{(\mathrm{b})}\omega_y\right) \\
\dot{v}_y &= \frac{1}{2}\left(\ \ u\ \ {}^{(\mathrm{b})}\omega_y - v_x\,{}^{(\mathrm{b})}\omega_z + v_z\,{}^{(\mathrm{b})}\omega_x\right) \\
\dot{v}_z &= \frac{1}{2}\left(\ \ u\ \ {}^{(\mathrm{b})}\omega_z + v_x\,{}^{(\mathrm{b})}\omega_y - v_y\,{}^{(\mathrm{b})}\omega_x\right)
\end{aligned}$$

A.2.3 Angular Acceleration

It can also be useful to differentiate w.r.t. time the relation between the time derivative of the angular rate and the acceleration of the chosen representation. This relation is used in flexible dynamics, where the angular deflection is represented by a set of Euler angles. The time derivative of matrix $R_{\boldsymbol{\Psi}}$ is then required when deriving $\dot{\mathbf{A}}^{\mathrm{f}\,(1)}_{i,i-1}$ in (5.50b). Using (A.7), the time derivative of the angular rate is derived by:

$$\dot{\boldsymbol{\omega}}_{\mathrm{b}} = \dot{R}_{\boldsymbol{\Psi}_{\mathrm{b}}}\,\dot{\boldsymbol{\Psi}}_{\mathrm{b}} + R_{\boldsymbol{\Psi}_{\mathrm{b}}}\,\ddot{\boldsymbol{\Psi}}_{\mathrm{b}}$$

where the only unknown term at this point is $\dot{R}_{\boldsymbol{\Psi}_{\mathrm{b}}}$.

Coming back to the expression (A.8) of the transformation for an angular rate expressed in inertial frame, its time derivative is obtained by:

$$\begin{aligned}
\dot{R}_{\boldsymbol{\Psi}_{\mathrm{b}}} &= \frac{\mathrm{d}R_{\boldsymbol{\Psi}_{\mathrm{b}}}}{\mathrm{d}t} \\
&= \dot{\Psi}_x\,\frac{\partial R_{\boldsymbol{\Psi}_{\mathrm{b}}}}{\partial \Psi_x} + \dot{\Psi}_y\,\frac{\partial R_{\boldsymbol{\Psi}_{\mathrm{b}}}}{\partial \Psi_y} + \dot{\Psi}_z\,\frac{\partial R_{\boldsymbol{\Psi}_{\mathrm{b}}}}{\partial \Psi_z}
\end{aligned}$$

Performing the derivation w.r.t. each Euler angle, the final expression is given as follows for ${}^{(\mathrm{I})}\boldsymbol{\omega}_{\mathrm{b}}$ expressed in the inertial frame:

$$\dot{R}_{\boldsymbol{\Psi}_{\mathrm{b}}} = \dot{\Psi}_y \begin{bmatrix} -c_z\,s_y & 0 & 0 \\ -s_z\,s_y & 0 & 0 \\ -\,c_y & 0 & 0 \end{bmatrix} + \dot{\Psi}_z \begin{bmatrix} -s_z\,c_y & -c_z & 0 \\ c_z\,c_y & -s_z & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{(A.12)}$$

while its equivalent for ${}^{(\mathrm{b})}\boldsymbol{\omega}_{\mathrm{b}}$ expressed in the local frame starts from (A.9) and reads:

$$\dot{R}_{\boldsymbol{\Psi}_{\mathrm{b}}} = \dot{\Psi}_x \begin{bmatrix} 0 & 0 & 0 \\ 0 & -s_x & c_x\, c_y \\ 0 & -c_x & -s_x\, c_y \end{bmatrix} + \dot{\Psi}_y \begin{bmatrix} 0 & 0 & -\,c_y \\ 0 & 0 & -s_x\, s_y \\ 0 & 0 & -c_x\, s_y \end{bmatrix} \tag{A.13}$$

A.3 Rigid Dynamics Algorithms

This section gathers the dedicated algorithms used to compute recursively the mass matrix, the vector of Coriolis and centrifugal forces, and the forward dynamics based on the DeNOC approach. The first part quickly covers the derivation of the generalized efforts at a given joint by using the wrench applied at its base. The resulting relation in (A.17) will then be used in the dynamic algorithms.

A.3.1 Generalized Efforts Computation

Reminder: An alternative proof is proposed to demonstrate that the generalized efforts at a joint can be expressed by the relation (4.42) and (5.54), as follows:

$$\boldsymbol{\tau}_i = \mathbf{P}_i^{\top}\, \boldsymbol{w}_i \tag{A.14}$$

Actually, this last relation represents the joint constraint from the efforts viewpoint. An alternative way is proposed by coming back to the dual relation on the velocity constraint in (4.32), written as:

$$\mathbf{t} = \mathbf{N}_l\, \mathbf{N}_d\, \dot{\mathbf{q}} \tag{A.15}$$

When inspecting the structure of the lower triangular matrix $\mathbf{N}_l$, one obtains the explicit dependency of any twist with respect to the generalized coordinates. With the convention that $\mathbf{A}_{i,i} = \mathbf{I}_6$, the summation reads as follows:

$$\mathbf{t}_k = \sum_{j=1}^{k} (\mathbf{A}_{k,j}\, \mathbf{P}_j\, \dot{q}_j) \tag{A.16}$$

This expression is now inserted into the definition of the generalized effort in (4.18), extended for the multi-body case and for the presence of external efforts $\{\boldsymbol{w}_{ext,k};\, k = 1 \ldots n\}$ (i.e., applied at the segment bases, at O_k). By merely summing over the number of segments and

using the fact that $\mathbf{A}_{i,j} = \mathbf{0}_{6\times 6}$ for $i < j$, it provides:

$$\begin{aligned}
\boldsymbol{Q}_{q_i} &= \sum_{k=1}^{n} \left(\left(\frac{\partial}{\partial \dot{q}_i} \mathbf{t}_{\mathrm{o}_k} \right)^{\top} (\boldsymbol{w}_{\mathrm{o}_k} + \boldsymbol{w}_{ext,k}) + \left(\frac{\partial}{\partial \dot{q}_i} \mathbf{t}_{\mathrm{e}_k} \right)^{\top} \boldsymbol{w}_{\mathrm{e}_k} \right) \\
&= \sum_{k=1}^{n} \left(\left(\frac{\partial}{\partial \dot{q}_i} \mathbf{t}_{\mathrm{o}_k} \right)^{\top} (\boldsymbol{w}_{\mathrm{o}_k} + \boldsymbol{w}_{ext,k}) + \left(\frac{\partial}{\partial \dot{q}_i} \mathbf{A}_{\mathrm{e}_k,\mathrm{o}_k} \mathbf{t}_{\mathrm{o}_k} \right)^{\top} \boldsymbol{w}_{\mathrm{e}_k} \right) \\
&= \sum_{k=1}^{n} \left(\left(\frac{\partial}{\partial \dot{q}_i} \mathbf{t}_{k} \right)^{\top} (\boldsymbol{w}_{\mathrm{o}_k} + \boldsymbol{w}_{ext,k}) - \left(\frac{\partial}{\partial \dot{q}_i} \mathbf{t}_{k} \right)^{\top} \mathbf{A}_{k+1,k}^{\top} \boldsymbol{w}_{k+1} \right)
\end{aligned}$$

where it is recalled that $\mathbf{A}_{k+1,k} \triangleq \mathbf{A}_{\mathrm{e}_k,\mathrm{o}_k}$. Considering now the twist expression in (A.16), it provides:

$$\boldsymbol{Q}_{q_i} = \sum_{k=1}^{n} \left(\frac{\partial}{\partial \dot{q}_i} \sum_{j=1}^{k} (\mathbf{A}_{k,j}\, \mathbf{P}_j\, \dot{q}_j) \right)^{\top} \left(\boldsymbol{w}_k + \boldsymbol{w}_{ext,k} - \mathbf{A}_{k+1,k}^{\top} \boldsymbol{w}_{k+1} \right)$$

The derivative w.r.t. $\dot{q}_i$ is straightforward since only the i^{th} term depends upon this variable, provided that $k \geq i$, otherwise the result is zero. Therefore the first sum is reduced to $k \geq i$, and reads as follows:

$$\begin{aligned}
\boldsymbol{Q}_{q_i} &= \sum_{k=i}^{n} (\mathbf{A}_{k,i}\, \mathbf{P}_i)^{\top} \left(\boldsymbol{w}_k + \boldsymbol{w}_{ext,k} - \mathbf{A}_{k+1,k}^{\top} \boldsymbol{w}_{k+1} \right) \\
&= \mathbf{P}_i^{\top} \left(\sum_{k=i}^{n} \mathbf{A}_{k,i}^{\top} \boldsymbol{w}_k + \sum_{k=i}^{n} \mathbf{A}_{k,i}^{\top} \boldsymbol{w}_{ext,k} - \sum_{k=i}^{n} \mathbf{A}_{k+1,i}^{\top} \boldsymbol{w}_{k+1} \right) \\
&= \mathbf{P}_i^{\top} \left(\sum_{k=i}^{n} \mathbf{A}_{k,i}^{\top} \boldsymbol{w}_k - \sum_{k'=i+1}^{n+1} \mathbf{A}_{k',i}^{\top} \boldsymbol{w}_{k'} + \sum_{k=i}^{n} \mathbf{A}_{k,i}^{\top} \boldsymbol{w}_{ext,k} \right) \\
&= \mathbf{P}_i^{\top} \left(\underbrace{\mathbf{A}_{i,i}^{\top}}_{=\mathbf{I}_6} \boldsymbol{w}_i - \mathbf{A}_{n+1,i}^{\top} \boldsymbol{w}_{n+1} + \sum_{k=i}^{n} \mathbf{A}_{k,i}^{\top} \boldsymbol{w}_{ext,k} \right)
\end{aligned}$$

This result can now be summarized as:

$$\boldsymbol{Q}_{q_i} = \boldsymbol{\tau}_i + \boldsymbol{\tau}_{i,\mathrm{E}} + \boldsymbol{\tau}_{i,ext} \tag{A.17}$$

with

$$\boldsymbol{\tau}_i = \mathbf{P}_i^{\top}\, \boldsymbol{w}_i \tag{A.18a}$$

$$\boldsymbol{\tau}_{i,\mathrm{E}} = \mathbf{P}_i^{\top}\, \mathbf{A}_{n+1,i}^{\top}\, \boldsymbol{w}_{n+1} \tag{A.18b}$$

$$\boldsymbol{\tau}_{i,ext} = \mathbf{P}_i^{\top} \sum_{k=i}^{n} \left(\mathbf{A}_{k,i}^{\top}\, \boldsymbol{w}_{ext,k} \right) \tag{A.18c}$$

The i^{th} generalized effort is thus split between the joint torque/force with $\boldsymbol{\tau}_i$, the effort induced by a wrench applied at the end-effector with $\boldsymbol{\tau}_{i,\mathrm{E}}$, and the efforts introduced by more general external efforts applied on the upper joints only, with $\boldsymbol{\tau}_{i,ext}$. It is worth noticing that the matrix multiplying the end-effector wrench in (A.18b) actually matches the transposed of the i^{th} column of the Jacobian matrix given in (4.33), which is confirmed by the term $\mathbf{J}_{\mathrm{E}}^{\top}\,\boldsymbol{w}_{\mathrm{E}}$ found in the closed-form equation of dynamics in (4.54).

A.3.2 Mass Matrix Algorithm

As mentioned earlier, a recursive method is already proposed to compute column by column the mass matrix of a manipulator using the inverse dynamics algorithms (Walker and Orin, 1982). Nevertheless, it appears to be more efficient to derive a dedicated algorithm to compute this mass matrix by taking advantage of its recursive structure. The next computations are closely related to the notion of *Composite Rigid Body* introduced in (Featherstone, 2008). In the scope of the DeNOC approach, the following algorithm is based on the work of S. K. Saha in (Saha, 1997, 1999).

Reminder: The mass matrix is obtained explicitly using the DeNOC approach in (4.55). The idea of the algorithm presented in this section is to compute efficiently this matrix product, by taking into account the block structure of the matrices instead of performing the raw product.

$$\mathbf{D}(\mathbf{q}) = \mathbf{N}_d^{\top}\,\mathbf{N}_l^{\top}\;[\mathbf{M}]\;\mathbf{N}_l\,\mathbf{N}_d \tag{A.19}$$

where general notation are used to stay valid for the rigid and flexible cases. The twist-propagation matrices $\mathbf{A}_{i,j}$ are stored in an lower triangular fashion in $\mathbf{N}_l$, while the joint-rate-propagation matrices $\mathbf{P}_i$ are diagonally stored in $\mathbf{N}_d$. Their expression is recalled in the general case of a payload, since its mass matrix will serve as initial condition in the algorithm.

$$\mathbf{N}_l = \left[\begin{array}{ccccc} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{2,1} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{3,1} & \mathbf{A}_{3,2} & \mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \cdots & \mathbf{0} \\ \mathbf{A}_{n,1} & \mathbf{A}_{n,2} & \cdots & \mathbf{A}_{n,n-1} & \mathbf{I} \\ \hline \mathbf{A}_{n+1,1} & \mathbf{A}_{n+1,2} & \cdots & \mathbf{A}_{n+1,n-1} & \mathbf{A}_{n+1,n} \end{array}\right] \tag{A.20}$$

$$\mathbf{N}_d = \begin{bmatrix} \mathbf{P}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_3 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_n \end{bmatrix} \tag{A.21}$$

It is also recalled that the twist-propagation matrices satisfy the following property of transitivity: $\mathbf{A}_{i,j} = \mathbf{A}_{i,k}\,\mathbf{A}_{k,j}$, and as a consequence, $\mathbf{A}_{i,i} = \mathbf{I}_6$.

By using the lower-triangular shape of $\mathbf{N}_l$ with the non-zero elements for $i \geq j$, the algorithm to recursively compute $\mathbf{D}$ in (Saha, 1997) is straightforward to prove. The complete algorithm is given for sake of completeness in Algorithm A.1, and is also valid for the flexible manipulators, as long as matrices $\mathbf{A}_{i,j}$ and $\mathbf{P}_i$ are updated with the flexible DoF.

Firstly, the inner product in (A.19) is denoted: $\widehat{\mathbf{M}} \triangleq \mathbf{N}_l^\top\,[\mathbf{M}]\,\mathbf{N}_l$. Applying the classic rules of matrix product by:

$$(A \times B)_{ij} = \sum_{k=1}^{n} A_{i,k}\,B_{k,j}$$

the terms of $\widehat{\mathbf{M}}$ are derived as follows:

$$\widehat{\mathbf{M}}_{i,j} = \sum_{k=1}^{n+1} \left(\mathbf{N}_l^\top\right)_{i,k} \left([\mathbf{M}]\,\mathbf{N}_l\right)_{k,j} = \sum_{k=1}^{n+1} \mathbf{A}_{k,i}^\top\,\mathbf{M}_k\,\mathbf{A}_{k,j} = \sum_{k=max(i,j)}^{n+1} \mathbf{A}_{k,i}^\top\,\mathbf{M}_k\,\mathbf{A}_{k,j}$$

When focusing on the diagonal terms, the lower right-hand sub-block of $\widehat{\mathbf{M}}$ is obtained by:

$$\widehat{\mathbf{M}}_{n,n} = \mathbf{M}_n + \mathbf{A}_{n+1,n}^\top\,\mathbf{M}_{n+1}\,\mathbf{A}_{n+1,n}$$

and for the remaining ones, an inward recursion appears:

$$\begin{aligned} \widehat{\mathbf{M}}_{i,i} &= \sum_{k=i}^{n} \mathbf{A}_{k,i}^\top\,\mathbf{M}_k\,\mathbf{A}_{k,i} \\ &= \mathbf{M}_i + \sum_{k=i+1}^{n} \mathbf{A}_{k,i}^\top\,\mathbf{M}_k\,\mathbf{A}_{k,i} \\ &= \mathbf{M}_i + \mathbf{A}_{i+1,i}^\top \left(\sum_{k=i+1}^{n} \mathbf{A}_{k,i+1}^\top\,\mathbf{M}_k\,\mathbf{A}_{k,i+1}\right) \mathbf{A}_{i+1,i} \\ &= \mathbf{M}_i + \mathbf{A}_{i+1,i}^\top\,\widehat{\mathbf{M}}_{i+1}\,\mathbf{A}_{i+1,i} \end{aligned}$$

For all the off-diagonal terms, the general expression can be re-written as a function of the

$\widehat{\mathbf{M}}_{i,i}$:

$$\begin{cases} \forall i \geq j, & \widehat{\mathbf{M}}_{i,j} = \sum_{k=i}^{n+1} \mathbf{A}_{k,i}^{\top} \mathbf{M}_k \mathbf{A}_{k,j} = \widehat{\mathbf{M}}_{i,i} \mathbf{A}_{i,j} \\ \forall i \leq j, & \widehat{\mathbf{M}}_{i,j} = \sum_{k=j}^{n+1} \mathbf{A}_{k,i}^{\top} \mathbf{M}_k \mathbf{A}_{k,j} = \mathbf{A}_{j,i}^{\top} \widehat{\mathbf{M}}_{j,j} \end{cases}$$

Denoting $\widehat{\mathbf{M}}_i \triangleq \widehat{\mathbf{M}}_{i,i}$ to alleviate the notations, the inner product is described recursively using the set of $\{\widehat{\mathbf{M}}_i;\ i = 1 \dots n+1\}$ matrices:

$$\widehat{\mathbf{M}} = \begin{bmatrix} \widehat{\mathbf{M}}_1 & \mathbf{A}_{2,1}^{\top} \widehat{\mathbf{M}}_2 & \cdots & \mathbf{A}_{n,1}^{\top} \widehat{\mathbf{M}}_n \\ \widehat{\mathbf{M}}_2 \mathbf{A}_{2,1} & \widehat{\mathbf{M}}_2 & \cdots & \mathbf{A}_{n,2}^{\top} \widehat{\mathbf{M}}_n \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{\mathbf{M}}_n \mathbf{A}_{n,1} & \widehat{\mathbf{M}}_n \mathbf{A}_{n,2} & \cdots & \widehat{\mathbf{M}}_n \end{bmatrix}$$

And the final mass matrix is derived by multiplying on the right and on the left hand side by, respectively, $\mathbf{N}_d$ and its transpose. It yields the algorithm in Algorithm A.1, summarized as follows:

$$\widehat{\mathbf{M}}_{n+1} = \mathbf{M}_{n+1} \tag{A.22a}$$

$$\forall i = n \dots 1, \quad \widehat{\mathbf{M}}_i = \mathbf{M}_i + \mathbf{A}_{i+1,i}^{\top} \widehat{\mathbf{M}}_{i+1} \mathbf{A}_{i+1,i} \tag{A.22b}$$

$$\forall i \geq j, \quad \mathbf{D}(i,j) = \mathbf{P}_i^{\top} \widehat{\mathbf{M}}_i \mathbf{A}_{i,j} \mathbf{P}_j \tag{A.22c}$$

$$\forall i \leq j, \quad \mathbf{D}(i,j) = \mathbf{D}(j,i)^{\top} \tag{A.22d}$$

Algorithm A.1: Recursive computation of the mass matrix of a rigid manipulator

Function : $\mathbf{D} = \texttt{MassMat}(\mathbf{q})$

Input : $\mathbf{q}$

Output : $\mathbf{D}$

Data : $\left\{ {}^{i-1}R_i^{\mathrm{r/f}}\,,\ \mathbf{A}_{i,i-1}^{\mathrm{r/f}}\,,\ \mathbf{P}_i\,,\ \mathbf{M}_i\ ;\ i = 1 \dots n+1 \right\}$

$\texttt{Initialize}\ \left(\widehat{\mathbf{M}}_{n+1} = \mathbf{M}_{n+1} = \mathbf{M}_{payload} \right)$

for $i = n \dots 1$ **do** // Recursive computation of $\widehat{\mathbf{M}}_i$

  ${}^{i+1}R_i = {}^{i}R_{i+1}^{\mathrm{r}}(q_{i+1})^\top$ or ${}^{i}R_{i+1}^{\mathrm{f}}(q_{i+1}, \boldsymbol{\delta}_i)^\top$ (4.22) and (5.35)

  ${}^{i+1}\boldsymbol{R}_i = \mathrm{diag}({}^{i+1}R_i, {}^{i+1}R_i)$ (4.31)

  $\mathbf{A}_{i+1,i} = {}^{i+1}\boldsymbol{R}_i\ \mathbf{A}_{i+1,i}^{\mathrm{r/f}}$ (4.30a) and (5.45a)

  $\widehat{\mathbf{M}}_i = \mathbf{M}_i + \mathbf{A}_{i+1,i}^\top\ \widehat{\mathbf{M}}_{i+1}\ \mathbf{A}_{i+1,i}$ (A.22b)

  $\mathbf{A}_{i,i} = \mathbf{I}_6$

  for $j = i \dots 1$ **do** // Symmetric terms of $\mathbf{N}_d^\top\ \widehat{\mathbf{M}}\ \mathbf{N}_d$

    $\mathbf{D}(i,j) = \mathbf{P}_i^\top\ \widehat{\mathbf{M}}_i\ \mathbf{A}_{i,j}\ \mathbf{P}_j$ (A.22c)

    $\mathbf{D}(j,i) = \mathbf{D}(i,j)^\top$ (A.22d)

    ${}^{j}R_{j-1} = {}^{j-1}R_j^{\mathrm{r}}(q_j)^\top$ or ${}^{j-1}R_j^{\mathrm{f}}(q_j, \boldsymbol{\delta}_{j-1})^\top$ // Update of $\mathbf{A}_{i,j}$

    ${}^{j}\boldsymbol{R}_{j-1} = \mathrm{diag}({}^{j}R_{j-1}, {}^{j}R_{j-1})$

    $\mathbf{A}_{i,j-1} = \mathbf{A}_{i,j}\ \left({}^{j}\boldsymbol{R}_{j-1}\ \mathbf{A}_{j,j-1}^{\mathrm{r/f}} \right)$

  end

end

---

return D

A.3.3 Coriolis and Centrifugal Matrix Algorithm

In the same way, the computation of the Coriolis and centrifugal vector can be performed recursively by using the decoupled form of $\mathbf{N}$ and its time derivative in (4.52) into its matrix expression in (4.56). Based on (Saha, 1999), this approach is built on the products between triangular and block-diagonal matrices.

Reminder: The result obtained in (4.56) is recalled here to highlight the structure of the Coriolis and centrifugal matrix.

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{N}^{\top}\,\mathbf{M}\,\dot{\mathbf{N}} + \mathbf{N}^{\top}\,[\boldsymbol{\Omega}]\,[\mathbf{M}]\,[\mathbf{E}_v]\,\mathbf{N} \tag{A.23}$$

where the general notation are still used since the relation is valid for both rigid and flexible cases.

Based on this expression, a recursive algorithm is derived to compute the whole matrix using the decoupled form of $\mathbf{N} = \mathbf{N}_l\,\mathbf{N}_d$. In order to develop an algorithm still valid for the flexible manipulators, the following notation is introduced:

$$\dot{\mathbf{M}}_{\mathrm{r}_i} \triangleq \boldsymbol{\Omega}_i\,\mathbf{M}_{\mathrm{r}_i}\,\mathbf{E}_v \tag{A.24}$$

since it is easily replaceable in the flexible case by:

$$\dot{\mathbf{M}}_{\mathrm{f}_i} \triangleq \left(\mathring{\mathbf{M}}_{\mathrm{f}_i} + \boldsymbol{\Omega}\,\mathbf{M}_{\mathrm{f}_i} + \tilde{\mathbf{M}}_{\mathrm{f}_i}\right)\mathbf{E}_v \tag{A.25}$$

With this notation, the diagonal form is denoted by $\left[\dot{\mathbf{M}}\right] = \mathrm{diag}(\dot{\mathbf{M}}_i,\, i = 1\cdots n+1)$. Developing then $\mathbf{N}$ with $\mathbf{N}_l$ and $\mathbf{N}_d$ into (A.23), the Coriolis and centrifugal matrix is given by:

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{N}_d^{\top}\,\left(\mathbf{N}_l^{\top}\,[\mathbf{M}]\,\dot{\mathbf{N}}_l + \mathbf{N}_l^{\top}\,[\mathbf{M}]\,\mathbf{N}_l\,[\boldsymbol{\Omega}] + \mathbf{N}_l^{\top}\,\left[\dot{\mathbf{M}}\right]\,\mathbf{N}_l\right)\mathbf{N}_d \tag{A.26}$$

where it must be recalled that $\dot{\mathbf{N}}_d = [\boldsymbol{\Omega}]\,\mathbf{N}_d$ from (4.52). In this expression of $\mathbf{C}$, two terms can be computed as $\widehat{\mathbf{M}}$ was:

- $\widehat{\mathbf{M}}\,\boldsymbol{\Omega}$: by multiplying the i^{th} block-column of $\widehat{\mathbf{M}}$ by $\boldsymbol{\Omega}_i$;
- $\hat{\dot{\mathbf{M}}} = \mathbf{N}_l^{\top}\left[\dot{\mathbf{M}}\right]\mathbf{N}_l$: by replacing $\mathbf{M}_i$ by $\dot{\mathbf{M}}_i$ in Algorithm A.1.

For the remaining term $\mathbf{N}_l^\top [\mathbf{M}] \dot{\mathbf{N}}_l$, a different algorithm must be used. The matrix $\mathbf{H}$ is introduced to denote :

$$
\begin{aligned}
\mathbf{H} &\triangleq \left(\mathbf{N}_l^\top [\mathbf{M}]\right) \dot{\mathbf{N}}_l \\
&= \left[\begin{array}{cccc|c}
\mathbf{M}_1 & \mathbf{A}_{2,1}^\top \mathbf{M}_2 & \cdots & \mathbf{A}_{n,1}^\top \mathbf{M}_n & \mathbf{A}_{n+1,1}^\top \mathbf{M}_{n+1} \\
\mathbf{0} & \mathbf{M}_2 & \cdots & \mathbf{A}_{n,2}^\top \mathbf{M}_n & \mathbf{A}_{n+1,2}^\top \mathbf{M}_{n+1} \\
\vdots & \ddots & \ddots & \vdots & \vdots \\
\mathbf{0} & \cdots & \mathbf{0} & \mathbf{M}_n & \mathbf{M}_{n+1}
\end{array}\right]
\left[\begin{array}{cccc}
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\dot{\mathbf{A}}_{2,1} & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \ddots & \ddots & \mathbf{0} \\
\dot{\mathbf{A}}_{n,1} & \cdots & \dot{\mathbf{A}}_{n,n-1} & \mathbf{0} \\
\hline
\dot{\mathbf{A}}_{n+1,1} & \cdots & \dot{\mathbf{A}}_{n+1,n-1} & \dot{\mathbf{A}}_{n+1,n}
\end{array}\right]
\end{aligned}
$$

Developing the matrix product and using the triangular shape to obtain that:

$$
\forall k < i,\ (\mathbf{N}_l^\top [\mathbf{M}])_{i,k} = 0 \qquad \text{and} \qquad \forall k \leq j,\ (\dot{\mathbf{N}}_l)_{k,j} = 0
$$

the terms of $\mathbf{H}$ are computed as follows, $\forall (i,j) \in \{1 \dots n\}$:

$$
\begin{aligned}
\mathbf{H}_{i,j} &= \sum_{k=max(i,j+1)}^{n+1} \mathbf{A}_{k,i}^\top \mathbf{M}_k \dot{\mathbf{A}}_{k,j} \\
&= \begin{cases}
\forall j < i, & \sum_{k=i}^{n+1} \mathbf{A}_{k,i}^\top \mathbf{M}_k \left(\dot{\mathbf{A}}_{k,i} \mathbf{A}_{i,j} + \mathbf{A}_{k,i} \dot{\mathbf{A}}_{i,j}\right) \\
\forall j \geq i & \sum_{k=j+1}^{n+1} \left(\mathbf{A}_{j,i}^\top \mathbf{A}_{k,j}^\top\right) \mathbf{M}_k \dot{\mathbf{A}}_{k,j}
\end{cases} \\
&= \begin{cases}
\forall j < i, & \left(\sum_{k=i}^{n+1} \mathbf{A}_{k,i}^\top \mathbf{M}_k \dot{\mathbf{A}}_{k,i}\right) \mathbf{A}_{i,j} + \left(\sum_{k=i}^{n+1} \mathbf{A}_{k,i}^\top \mathbf{M}_k \mathbf{A}_{k,i}\right) \dot{\mathbf{A}}_{i,j} \\
\forall j \geq i & \mathbf{A}_{j,i}^\top \left(\sum_{k=j+1}^{n+1} \mathbf{A}_{k,j}^\top \mathbf{M}_k \dot{\mathbf{A}}_{k,j}\right)
\end{cases} \\
&= \begin{cases}
\forall j < i, & \mathbf{H}_{i,i} \mathbf{A}_{i,j} + \widehat{\mathbf{M}}_i \dot{\mathbf{A}}_{i,j} \\
\forall j \geq i & \mathbf{A}_{j,i}^\top \mathbf{H}_{j,j}
\end{cases}
\end{aligned}
$$

where the properties of the matrices $\mathbf{A}_{i,j}$ are used, as an example: $\mathbf{A}_{i,i} = \mathbf{I}_6 \Longrightarrow \dot{\mathbf{A}}_{i,i} = \mathbf{0}_{6\times 6}$. Similar to the computation of $\widehat{\mathbf{M}}$, the notations are alleviated by denoting $\mathbf{H}_i \triangleq \mathbf{H}_{i,i}$, which obeys the following recursive relation:

$$
\begin{aligned}
\mathbf{H}_i &= \sum_{k=i+1}^{n+1} \left(\mathbf{A}_{i+1,i}^\top \mathbf{A}_{k,i+1}^\top\right) \mathbf{M}_k \left(\dot{\mathbf{A}}_{k,i+1} \mathbf{A}_{i+1,i} + \mathbf{A}_{k,i+1} \dot{\mathbf{A}}_{i+1,i}\right) \\
&= \mathbf{A}_{i+1,i}^\top \left(\sum_{k=i+1}^{n+1} \mathbf{A}_{k,i+1}^\top \mathbf{M}_k \left(\dot{\mathbf{A}}_{k,i+1} \mathbf{A}_{i+1,i} + \mathbf{A}_{k,i+1} \dot{\mathbf{A}}_{i+1,i}\right)\right) \\
&= \mathbf{A}_{i+1,i}^\top \left(\mathbf{H}_{i+1} \mathbf{A}_{i+1,i} + \widehat{\mathbf{M}}_{i+1} \dot{\mathbf{A}}_{i+1,i}\right)
\end{aligned}
$$

The resulting recursion on $\mathbf{H}_i$ matrices is summarized as follows, with an initialization done on $\mathbf{H}_{n+1}$[1] instead of $\mathbf{H}_n$:

$$\left\{\begin{array}{lrcl} & \mathbf{H}_{n+1} & = & \mathbf{0}_{6\times 6} \\ \forall i = n \dots 1, & \mathbf{H}_i & = & \mathbf{A}_{i+1,i}^\top \left(\mathbf{H}_{i+1}\mathbf{A}_{i+1,i} + \widehat{\mathbf{M}}_{i+1}\dot{\mathbf{A}}_{i+1,i} \right) \end{array}\right.$$

Eventually, the global expression of $\mathbf{H}$ is given as follows, with an arrow-shaped matrix in the second term:

$$\mathbf{H} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \widehat{\mathbf{M}}_2\dot{\mathbf{A}}_{2,1} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \widehat{\mathbf{M}}_n\dot{\mathbf{A}}_{n,1} & \cdots & \widehat{\mathbf{M}}_n\dot{\mathbf{A}}_{n,n-1} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{H}_1 & \mathbf{A}_{2,1}^\top\mathbf{H}_2 & \cdots & \mathbf{A}_{n,1}^\top\mathbf{H}_n \\ \mathbf{H}_2\mathbf{A}_{2,1} & \mathbf{H}_2 & \cdots & \mathbf{A}_{n,2}^\top\mathbf{H}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_n\mathbf{A}_{n,1} & \mathbf{H}_n\mathbf{A}_{n,2} & \cdots & \mathbf{H}_n \end{bmatrix} \tag{A.27}$$

This recursive expression of $\mathbf{H}_i$ results in a numerically more efficient algorithm to compute $\mathbf{H}$. Indeed, only the two sets of n matrices $\left\{\widehat{\mathbf{M}}_i \; ; \; i = 1 \dots n\right\}$ and $\{\mathbf{H}_i \; ; \; i = 1 \dots n\}$ need to be known to compute $\mathbf{H}$, instead of the whole triangular part $\{\mathbf{H}_{i,j} \; ; \; i = n \dots 1, j = i-1 \dots 1\}$ as proposed in (Saha, 1999).

Using the previous expression of $\mathbf{C}_\mathrm{r}$, a formula similar to (29) in (Saha, 1999) is derived, starting from its lower right to its upper left corner, and proceeding line by line:

$$\forall i = n \dots 1, \qquad \widehat{\mathbf{M}}_i = \mathbf{M}_i + \mathbf{A}_{i+1,i}^\top \widehat{\mathbf{M}}_{i+1} \mathbf{A}_{i+1,i} \tag{A.28a}$$

$$\forall i = n \dots 1, \qquad \dot{\widehat{\mathbf{M}}}_i = \dot{\mathbf{M}}_i + \mathbf{A}_{i+1,i}^\top \dot{\widehat{\mathbf{M}}}_{i+1} \mathbf{A}_{i+1,i} \tag{A.28b}$$

$$\forall i = n \dots 1, \qquad \mathbf{H}_i = \mathbf{A}_{i+1,i}^\top \left(\mathbf{H}_{i+1}\mathbf{A}_{i+1,i} + \widehat{\mathbf{M}}_{i+1}\dot{\mathbf{A}}_{i+1,i} \right) \tag{A.28c}$$

$$\forall i, \forall j < i, \qquad \mathbf{C}_\mathrm{r}(i,j) = \mathbf{P}_i^\top \left(\left(\mathbf{H}_i + \dot{\widehat{\mathbf{M}}}_i \right) \mathbf{A}_{i,j} + \widehat{\mathbf{M}}_i \left(\dot{\mathbf{A}}_{i,j} + \mathbf{A}_{i,j}\boldsymbol{\Omega}_j \right) \right) \mathbf{P}_j \tag{A.28d}$$

$$\forall i, \forall j \geq i, \qquad \mathbf{C}_\mathrm{r}(i,j) = \mathbf{P}_i^\top \mathbf{A}_{j,i}^\top \left(\mathbf{H}_j + \widehat{\mathbf{M}}_j\boldsymbol{\Omega}_j + \dot{\widehat{\mathbf{M}}}_j \right) \mathbf{P}_j \tag{A.28e}$$

The complete algorithm to compute the Coriolis and centrifugal terms is given in Algorithm A.2. One main advantage of this formulation of $\mathbf{h} = \mathbf{C}\,\dot{\mathbf{q}}$ is its use for control and especially for the linearization. Moreover, it provides an insight into the structure of the nonlinear terms, and how they affect the global dynamics.

[1] which is equivalent, by recursion, to: $\mathbf{H}_n = \mathbf{H}_{n,n} = \mathbf{A}_{n+1,n}^\top \mathbf{M}_{n+1} \dot{\mathbf{A}}_{n+1,n}$, obtained by evaluating the sum at $i = n$.

Algorithm A.2: Computation of the Coriolis and centrifugal vector based on matrix $\mathbf{C}_\mathrm{r}$

Function : $\mathbf{h}_\mathrm{r} = \texttt{CorMat}(\mathbf{q})$
Input : $\mathbf{q}$
Output : $\mathbf{h}$
Data : $\left\{ {}^{i-1}R_i^{\mathrm{r/f}}\,,\ \mathbf{A}_{i,i-1}^{\mathrm{r/f}}\,,\ \mathbf{P}_i\,,\ \mathbf{M}_i\ ;\ i = 1 \dots n+1 \right\}$

`Initialize` $\left(\mathbf{t}_0 = \mathbf{0}_{6\times1},\ \mathbf{A}_{1,0}^{\mathrm{r}} = \mathbf{I}_6,\ \dot{\mathbf{A}}_{1,0}^{\mathrm{r}} = \mathbf{0}_{6\times6}\right)$ // **KINETICS LOOP**

1 **for** $i = 1 \dots n+1$ **do** // Recursive computation and storage of $\mathbf{A}_{i,i-1}$, $\dot{\mathbf{A}}_{i,i-1}$, and $\boldsymbol{\Omega}_i$

Kinetics loop from `InvDynRig`
(only $\dot{\mathbf{t}}_i$ is not computed here)

end

`Initialize` $\left(\widehat{\mathbf{M}}_{n+1} = \mathbf{M}_{n+1},\ \dot{\widehat{\mathbf{M}}}_{n+1} = \boldsymbol{\Omega}_{n+1}\mathbf{M}_{n+1}\mathbf{E}_v,\ \mathbf{H}_{n+1} = \mathbf{0}_{6\times6}\right)$ // **DYNAMICS LOOP**

2 **for** $i = n \dots 1$ **do** // Recursive computation of $\widehat{\mathbf{M}}_i, \dot{\widehat{\mathbf{M}}}_i, \mathbf{H}_i$

3 $\quad \widehat{\mathbf{M}}_i = \mathbf{M}_i + \mathbf{A}_{i+1,i}^\top\, \widehat{\mathbf{M}}_{i+1}\, \mathbf{A}_{i+1,i}$ (A.28a)

4 $\quad \dot{\mathbf{M}}_i = \boldsymbol{\Omega}_i\, \mathbf{M}_i\, \mathbf{E}_v$ (A.24)

5 $\quad \dot{\widehat{\mathbf{M}}}_i = \dot{\mathbf{M}}_i + \mathbf{A}_{i+1,i}^\top\, \dot{\widehat{\mathbf{M}}}_{i+1}\, \mathbf{A}_{i+1,i}$ (A.28b)

6 $\quad \mathbf{H}_i = \mathbf{A}_{i+1,i}^\top \left(\widehat{\mathbf{M}}_{i+1}\, \dot{\mathbf{A}}_{i+1,i} + \mathbf{H}_{i+1}\, \mathbf{A}_{i+1,i}\right)$ (A.28c)

7 $\quad \mathbf{A}_{i,i} = \mathbf{I}_6$

8 $\quad \dot{\mathbf{A}}_{i,i} = \mathbf{0}_{6\times6}$

9 $\quad$ **for** $j = i-1 \dots 1$ **do** // Computation of the LHS of line $\mathbf{C}_\mathrm{r}(i,:)$

10 $\quad\quad \mathbf{A}_{i,j} = \mathbf{A}_{i,j+1}\, \mathbf{A}_{j+1,j}$

11 $\quad\quad \dot{\mathbf{A}}_{i,j} = \dot{\mathbf{A}}_{i,j+1}\, \mathbf{A}_{j+1,j} + \mathbf{A}_{i,j+1}\, \dot{\mathbf{A}}_{j+1,j}$

12 $\quad\quad \mathbf{C}_\mathrm{r}(i,j) = \mathbf{P}_i^\top \left(\left(\mathbf{H}_i + \dot{\widehat{\mathbf{M}}}_i \right) \mathbf{A}_{i,j} + \widehat{\mathbf{M}}_i \left(\dot{\mathbf{A}}_{i,j} + \mathbf{A}_{i,j}\boldsymbol{\Omega}_j \right) \right) \mathbf{P}_j$ (A.28d)

$\quad$ **end**

13 $\quad$ **for** $j = i \dots n$ **do** // Computation of RHS of line $\mathbf{C}_\mathrm{r}(i,:)$

14 $\quad\quad \mathbf{C}_\mathrm{r}(i,j) = \mathbf{P}_i^\top \mathbf{A}_{j,i}^\top \left(\mathbf{H}_j + \widehat{\mathbf{M}}_j\boldsymbol{\Omega}_j + \dot{\widehat{\mathbf{M}}}_j \right) \mathbf{P}_j$ (A.28e)

15 $\quad\quad \mathbf{A}_{j+1,i} = \mathbf{A}_{j+1,j}\, \mathbf{A}_{j,i}$

$\quad$ **end**

end

---

16 **return** $\mathbf{h} = \mathbf{C}\,\dot{\mathbf{q}}$

A.3.4 Forward Dynamics Algorithm

This section aims at deriving the analytical expression of the sub-matrices $\mathbf{\Delta}_i$ and $\mathbf{U}_{i,j}$ used to invert the whole mass matrix $\mathbf{D}$ following the RGE technique presented in (Saha, 1999; Mohan and Saha, 2007, 2009).

Reminder: The forward dynamics consists in computing the acceleration resulting from efforts $\boldsymbol{\tau}$ applied at the joints, and assuming a given configuration $(\mathbf{q}, \dot{\mathbf{q}})$ of the manipulator. When considering the closed-form equation, this task is actually performed by inverting the mass matrix :

$$\ddot{\mathbf{q}} = \mathbf{D}(\mathbf{q})^{-1} \left(\boldsymbol{\tau} - \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \right) \tag{A.29}$$

where the external efforts at end-effector have been canceled, by considering only a payload attached at this point. In addition, the Coriolis and centrifugal terms are denoted by $\mathbf{h}$ to encompass both rigid and flexible cases, where the stiffness term may appear.

The inversion of $\mathbf{D}$ is then performed by taking advantage of its recursive structure described in Algorithm A.1, and by using an algorithm developed by S. K. Saha in (Saha, 1997). This approach is based on the $\mathbf{U}\mathbf{\Delta}\mathbf{U}^\top$ decomposition of the symmetric mass matrix $\mathbf{D}$:

$$\exists \mathbf{U} \in \mathbb{R}^{n \times n},\ \mathbf{\Delta} \in \mathbb{R}^{n \times n} |\quad \mathbf{D} = \mathbf{U}\,\mathbf{\Delta}\,\mathbf{U}^\top$$

with $\mathbf{U}$ a block upper triangular matrix and $\mathbf{\Delta}$ a block diagonal matrix.

$$\mathbf{U} = \begin{bmatrix} \mathbf{I} & \mathbf{U}_{1,2} & \cdots & \mathbf{U}_{1,n} \\ \mathbf{0} & \mathbf{I} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{U}_{n-1,n} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{bmatrix} \quad \mathbf{\Delta} = \begin{bmatrix} \mathbf{\Delta}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{\Delta}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{\Delta}_n \end{bmatrix} \quad \mathbf{U}^\top = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{U}_{1,2}^\top & \mathbf{I} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{U}_{1,n}^\top & \cdots & \mathbf{U}_{n-1,n}^\top & \mathbf{I} \end{bmatrix}$$

Developing the product $\mathbf{U}\mathbf{\Delta}\mathbf{U}^\top$, one obtains:

$$\mathbf{U}\,\mathbf{\Delta}\,\mathbf{U}^\top = \begin{bmatrix} \mathbf{\Delta}_1 + \sum_{k=2}^{n} \mathbf{U}_{1,k}\mathbf{\Delta}_k\mathbf{U}_{1,k}^\top & \mathbf{U}_{1,2}\mathbf{\Delta}_2 + \sum_{k=3}^{n} \mathbf{U}_{1,k}\mathbf{\Delta}_k\mathbf{U}_{2,k}^\top & \cdots & \mathbf{U}_{1,n}\mathbf{\Delta}_n \\ \mathbf{\Delta}_2\mathbf{U}_{1,2}^\top + \sum_{k=3}^{n} \mathbf{U}_{2,k}\mathbf{\Delta}_k\mathbf{U}_{1,k}^\top & \mathbf{\Delta}_2 + \sum_{k=3}^{n} \mathbf{U}_{2,k}\mathbf{\Delta}_k\mathbf{U}_{2,k}^\top & \cdots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{U}_{n-1,n}\mathbf{\Delta}_n \\ \mathbf{\Delta}_n\mathbf{U}_{1,n}^\top & \cdots & \mathbf{\Delta}_n\mathbf{U}_{n-1,n}^\top & \mathbf{\Delta}_n \end{bmatrix}$$

While the analytical expression of $\mathbf{D}$, lying on (A.19), provided that:

$$\mathbf{D} = \mathbf{N}^\top \, [\mathbf{M}] \, \mathbf{N} = \begin{bmatrix} \mathbf{P}_1^\top \widehat{\mathbf{M}}_1 \mathbf{P}_1 & \mathbf{P}_1^\top \mathbf{A}_{2,1}^\top \widehat{\mathbf{M}}_2 \mathbf{P}_2 & \cdots & \mathbf{P}_1^\top \mathbf{A}_{n,1}^\top \widehat{\mathbf{M}}_n \mathbf{P}_n \\ \mathbf{P}_2^\top \widehat{\mathbf{M}}_2 \mathbf{A}_{2,1} \mathbf{P}_1 & \mathbf{P}_2^\top \widehat{\mathbf{M}}_2 \mathbf{P}_2 & \cdots & \mathbf{P}_2^\top \mathbf{A}_{n,2}^\top \widehat{\mathbf{M}}_n \mathbf{P}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_n^\top \widehat{\mathbf{M}}_n \mathbf{A}_{n,1} \mathbf{P}_1 & \mathbf{P}_n^\top \widehat{\mathbf{M}}_n \mathbf{A}_{n,2} \mathbf{P}_2 & \cdots & \mathbf{P}_n^\top \widehat{\mathbf{M}}_n \mathbf{P}_n \end{bmatrix}$$

N.B.: For a 3 segments example, the previous sums of matrices can be expanded to give an idea of the recursions arising in the expression of the global mass matrix. It yields:

$$\begin{bmatrix} \boldsymbol{\Delta}_1 + \mathbf{U}_{1,2} \boldsymbol{\Delta}_2 \mathbf{U}_{1,2}^\top + \mathbf{U}_{1,3} \boldsymbol{\Delta}_3 \mathbf{U}_{1,3}^\top & \mathbf{U}_{1,2} \boldsymbol{\Delta}_2 + \mathbf{U}_{1,3} \boldsymbol{\Delta}_3 \mathbf{U}_{2,3}^\top & \mathbf{U}_{1,3} \boldsymbol{\Delta}_3 \\ \boldsymbol{\Delta}_2 \mathbf{U}_{1,2}^\top + \mathbf{U}_{2,3} \boldsymbol{\Delta}_3 \mathbf{U}_{1,3}^\top & \boldsymbol{\Delta}_2 + \mathbf{U}_{2,3} \boldsymbol{\Delta}_3 \mathbf{U}_{3,2}^\top & \mathbf{U}_{2,3} \boldsymbol{\Delta}_3 \\ \boldsymbol{\Delta}_3 \mathbf{U}_{1,3}^\top & \boldsymbol{\Delta}_3 \mathbf{U}_{2,3}^\top & \boldsymbol{\Delta}_3 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{P}_1^\top \widehat{\mathbf{M}}_1 \mathbf{P}_1 & \mathbf{P}_1^\top \mathbf{A}_{2,1}^\top \widehat{\mathbf{M}}_2 \mathbf{P}_2 & \mathbf{P}_1^\top \mathbf{A}_{3,1}^\top \widehat{\mathbf{M}}_3 \mathbf{P}_3 \\ \mathbf{P}_2^\top \widehat{\mathbf{M}}_2 \mathbf{A}_{2,1} \mathbf{P}_1 & \mathbf{P}_2^\top \widehat{\mathbf{M}}_2 \mathbf{P}_2 & \mathbf{P}_2^\top \mathbf{A}_{3,2}^\top \widehat{\mathbf{M}}_3 \mathbf{P}_3 \\ \mathbf{P}_3^\top \widehat{\mathbf{M}}_3 \mathbf{A}_{3,1} \mathbf{P}_1 & \mathbf{P}_3^\top \widehat{\mathbf{M}}_3 \mathbf{A}_{3,2} \mathbf{P}_2 & \mathbf{P}_3^\top \widehat{\mathbf{M}}_3 \mathbf{P}_3 \end{bmatrix}$$

When focusing on the lower triangular part $\{(i,j); j \leq i\}$, the identity provides:

$$\mathbf{D}_{i,j} = \begin{cases} \boldsymbol{\Delta}_i \mathbf{U}_{j,i}^\top + \sum_{k=i+1}^{n} \mathbf{U}_{i,k} \boldsymbol{\Delta}_k \mathbf{U}_{j,k}^\top \\ \mathbf{P}_i^\top \widehat{\mathbf{M}}_i \mathbf{A}_{i,j} \mathbf{P}_j \end{cases} \tag{A.30}$$

Deriving the last row, the last sub-block of $\boldsymbol{\Delta}$ is obtained by:

$$\boldsymbol{\Delta}_n = \mathbf{P}_n^\top \, \widehat{\mathbf{M}}_n \, \mathbf{P}_n \tag{A.31}$$

and some unknowns $\mathbf{U}_{i,j}$ can already be solved by:

$$\forall j = 1 \ldots (n-1), \quad \mathbf{U}_{j,n} = \left(\mathbf{P}_j^\top \mathbf{A}_{n,j}^\top \widehat{\mathbf{M}}_n \mathbf{P}_n \right) \boldsymbol{\Delta}_n^{-1} \tag{A.32}$$

Applying the same kind of reasoning on the $(n-1)^{th}$ row, it yields:

$$\forall j = 1 \dots (n-2), \quad \boldsymbol{\Delta}_{n-1}\mathbf{U}_{j,n-1}^{\top} + \mathbf{U}_{n-1,n}\boldsymbol{\Delta}_{n}\mathbf{U}_{j,n}^{\top} = \mathbf{P}_{n-1}^{\top}\widehat{\mathbf{M}}_{n-1}\mathbf{A}_{n-1,j}\mathbf{P}_{j}$$

such that, resolving for $\mathbf{U}_{j,n-1}$:

$$\begin{aligned}
\mathbf{U}_{j,n-1} &= \left(\mathbf{P}_{j}^{\top}\mathbf{A}_{n-1,j}^{\top}\widehat{\mathbf{M}}_{n-1}\mathbf{P}_{n-1} - \mathbf{U}_{j,n}\boldsymbol{\Delta}_{n}\mathbf{U}_{n-1,n}^{\top}\right)\boldsymbol{\Delta}_{n-1}^{-1} \\
&= \left(\mathbf{P}_{j}^{\top}\mathbf{A}_{n-1,j}^{\top}\widehat{\mathbf{M}}_{n-1}\mathbf{P}_{n-1} - \left(\mathbf{P}_{j}^{\top}\mathbf{A}_{n,j}^{\top}\widehat{\mathbf{M}}_{n}\mathbf{P}_{n}\right)\boldsymbol{\Delta}_{n}^{-1}\boldsymbol{\Delta}_{n}\boldsymbol{\Delta}_{n}^{-1}\left(\mathbf{P}_{n}^{\top}\widehat{\mathbf{M}}_{n}\mathbf{A}_{n,n-1}\mathbf{P}_{n-1}\right)\right)\boldsymbol{\Delta}_{n-1}^{-1} \\
&= \mathbf{P}_{j}^{\top}\mathbf{A}_{n-1,j}^{\top}\left(\widehat{\mathbf{M}}_{n-1} - \mathbf{A}_{n,n-1}^{\top}\left(\widehat{\mathbf{M}}_{n}\mathbf{P}_{n}\boldsymbol{\Delta}_{n}^{-1}\mathbf{P}_{n}^{\top}\widehat{\mathbf{M}}_{n}\right)\mathbf{A}_{n,n-1}\right)\mathbf{P}_{n-1}\boldsymbol{\Delta}_{n-1}^{-1}
\end{aligned}$$

Developing then $\widehat{\mathbf{M}}_{n-1}$ using (A.22b), it yields:

$$\begin{aligned}
\mathbf{U}_{j,n-1} &= \mathbf{P}_{j}^{\top}\mathbf{A}_{n-1,j}^{\top}\left(\mathbf{M}_{n-1} + \mathbf{A}_{n,n-1}^{\top}\widehat{\mathbf{M}}_{n}\mathbf{A}_{n,n-1} - \mathbf{A}_{n,n-1}^{\top}\left(\widehat{\mathbf{M}}_{n}\mathbf{P}_{n}\boldsymbol{\Delta}_{n}^{-1}\mathbf{P}_{n}^{\top}\widehat{\mathbf{M}}_{n}\right)\mathbf{A}_{n,n-1}\right)\mathbf{P}_{n-1}\boldsymbol{\Delta}_{n-1}^{-1} \\
&= \mathbf{P}_{j}^{\top}\mathbf{A}_{n-1,j}^{\top}\left(\underbrace{\mathbf{M}_{n-1} + \mathbf{A}_{n,n-1}^{\top}\left(\widehat{\mathbf{M}}_{n} - \widehat{\mathbf{M}}_{n}\mathbf{P}_{n}\boldsymbol{\Delta}_{n}^{-1}\mathbf{P}_{n}^{\top}\widehat{\mathbf{M}}_{n}\right)\mathbf{A}_{n,n-1}}_{\triangleq\bar{\mathbf{M}}_{n-1}}\right)\mathbf{P}_{n-1}\boldsymbol{\Delta}_{n-1}^{-1}
\end{aligned}$$

From this result, it is expected that a similar formula can be derived for all matrices $\mathbf{U}_{i,j}$. It is assumed that:

$$\boxed{\mathbf{U}_{i,j} = \left(\mathbf{P}_{i}^{\top}\mathbf{A}_{j,i}^{\top}\bar{\mathbf{M}}_{j}\mathbf{P}_{j}\right)\boldsymbol{\Delta}_{j}^{-1}} \tag{A.33}$$

In this expression, it is worth mentioning that matrices $\bar{\mathbf{M}}_i$ are matching the *Articulated-Body-Inertia* terms presented in (Featherstone, 2008). They are equivalent in size with previous mass matrices $\mathbf{M}_i$ or $\widehat{\mathbf{M}}_i$, and must satisfy the following recursion $\forall i = n-1 \dots 1$:

$$\boxed{\begin{cases} \bar{\mathbf{M}}_{n+1} &= \widehat{\mathbf{M}}_{n+1}\,(=\mathbf{M}_{n+1}) \\ \bar{\mathbf{M}}_{i} &= \mathbf{M}_{i} + \mathbf{A}_{i+1,i}^{\top}\left(\bar{\mathbf{M}}_{i+1} - \boldsymbol{\varphi}_{i+1}\,\bar{\boldsymbol{\varphi}}_{i+1}^{\top}\right)\mathbf{A}_{i+1,i} \end{cases}} \tag{A.34}$$

with the intermediate variables $(\boldsymbol{\varphi}_i, \bar{\boldsymbol{\varphi}}_i)$, initialized by $\bar{\boldsymbol{\varphi}}_{n+1} = \boldsymbol{\varphi}_{n+1} = \mathbf{0}_{6\times 1}$, and defined by:

$$\forall i = n \dots 1, \quad \bar{\boldsymbol{\varphi}}_{i} = \bar{\mathbf{M}}_{i}\,\mathbf{P}_{i} \tag{A.35}$$

$$\forall i = n \dots 1, \quad \boldsymbol{\varphi}_{i} = \bar{\boldsymbol{\varphi}}_{i}\,\boldsymbol{\Delta}_{i}^{-1} = \bar{\mathbf{M}}_{i}\,\mathbf{P}_{i}\,\boldsymbol{\Delta}_{i}^{-1} \tag{A.36}$$

In parallel, the sub-blocks of $\boldsymbol{\Delta}$ are defined $\forall i = 1 \dots n$ by:

$$\boxed{\boldsymbol{\Delta}_{i} = \mathbf{P}_{i}^{\top}\bar{\mathbf{M}}_{i}\mathbf{P}_{i}} \tag{A.37}$$

PROOF:

The proof is led around the following definition:

$$\bar{\mathbf{M}}_j \triangleq \widehat{\mathbf{M}}_j - \sum_{k=j+1}^{n} \mathbf{A}_{k,j}^{\top} \boldsymbol{\varphi}_k \bar{\boldsymbol{\varphi}}_k^{\top} \mathbf{A}_{k,j} \tag{A.38}$$

Hypothesis of recursion: The previous recursive relations in (A.33), (A.34) and (A.37) are assumed to be true at step $j+1$ for the following sets of matrices:

$$\begin{aligned}
&\forall l = n \dots j+1, && \bar{\mathbf{M}}_l = \mathbf{M}_l + \mathbf{A}_{l+1,l}^{\top} \left(\bar{\mathbf{M}}_{l+1} - \boldsymbol{\varphi}_{l+1} \bar{\boldsymbol{\varphi}}_{l+1}^{\top} \right) \mathbf{A}_{l+1,l} \\
&\forall l = n \dots j+1, && \boldsymbol{\Delta}_l = \mathbf{P}_l^{\top} \bar{\mathbf{M}}_l \mathbf{P}_l \\
&\forall l = n \dots j+1, \forall i = l-1 \dots 1, && \mathbf{U}_{i,l} = \mathbf{P}_i^{\top} \mathbf{A}_{l,i}^{\top} \boldsymbol{\varphi}_l
\end{aligned}$$

Initial check at $j = n$: Regarding $\bar{\mathbf{M}}_n$, its own recursion is developed as follows with the initialization of $\bar{\boldsymbol{\varphi}}_{n+1}$ and $\boldsymbol{\varphi}_{n+1}$ in (A.35) and (A.36):

$$\bar{\mathbf{M}}_n = \mathbf{M}_n + \mathbf{A}_{n+1,n}^{\top} \widehat{\mathbf{M}}_{n+1} \mathbf{A}_{n+1,n} = \widehat{\mathbf{M}}_n$$

which satisfies the definition of $\bar{\mathbf{M}}_n$ in (A.38), since the sum vanishes for $j \geq n$. Thanks to this relation, the properties are satisfied for $\boldsymbol{\Delta}_n$ given in (A.31), and for $\{\mathbf{U}_{i,n};\ i = 1 \dots (n-1)\}$ in (A.32) by replacing $\widehat{\mathbf{M}}_n$ by $\bar{\mathbf{M}}_n$ in both.

Next iteration check: Assuming the hypothesis of recursion holds at $j+1$, the results are shown at the next step j. To do this, the equality in (A.30) is derived first for the diagonal terms $i = j$:

$$\boldsymbol{\Delta}_j + \sum_{k=j+1}^{n} \mathbf{U}_{j,k} \boldsymbol{\Delta}_k \mathbf{U}_{j,k}^{\top} = \mathbf{P}_j^{\top} \widehat{\mathbf{M}}_j \mathbf{P}_j$$

Using the hypothesis to develop the terms $\mathbf{U}_{j,k}$, it leads to:

$$\begin{aligned}
\boldsymbol{\Delta}_j &= \mathbf{P}_j^{\top} \widehat{\mathbf{M}}_j \mathbf{P}_j - \sum_{k=j+1}^{n} \mathbf{U}_{j,k} \boldsymbol{\Delta}_k \mathbf{U}_{j,k}^{\top} \\
&= \mathbf{P}_j^{\top} \widehat{\mathbf{M}}_j \mathbf{P}_j - \sum_{k=j+1}^{n} \mathbf{P}_j^{\top} \mathbf{A}_{k,j}^{\top} \bar{\mathbf{M}}_k \mathbf{P}_k \boldsymbol{\Delta}_k^{-1} \boldsymbol{\Delta}_k \boldsymbol{\Delta}_k^{-1} \mathbf{P}_k^{\top} \bar{\mathbf{M}}_k \mathbf{A}_{k,j} \mathbf{P}_j \\
&= \mathbf{P}_j^{\top} \left(\widehat{\mathbf{M}}_j - \left(\sum_{k=j+1}^{n} \mathbf{A}_{k,j}^{\top} \bar{\mathbf{M}}_k \mathbf{P}_k \boldsymbol{\Delta}_k^{-1} \mathbf{P}_k^{\top} \bar{\mathbf{M}}_k \mathbf{A}_{k,j} \right) \mathbf{A}_{j+1,j} \right) \mathbf{P}_j \\
&= \mathbf{P}_j^{\top} \left(\widehat{\mathbf{M}}_j - \sum_{k=j+1}^{n} \mathbf{A}_{k,j}^{\top} \boldsymbol{\varphi}_k \bar{\boldsymbol{\varphi}}_k^{\top} \mathbf{A}_{k,j} \right) \mathbf{P}_j
\end{aligned}$$

By definition of $\bar{\mathbf{M}}_j$, it yields that:

$$\boldsymbol{\Delta}_j = \mathbf{P}_j^\top \, \bar{\mathbf{M}}_j \, \mathbf{P}_j$$

which proves the recursion for $\boldsymbol{\Delta}$. In addition, when this expression $\bar{\mathbf{M}}_j$ is developed according to the recursion of $\widehat{\mathbf{M}}[j+1]$ itself, it leads to:

$$\begin{aligned}
\bar{\mathbf{M}}_j &= \mathbf{M}_j + \mathbf{A}_{j+1,j}^\top \widehat{\mathbf{M}}_{j+1} \mathbf{A}_{j+1,j} - \sum_{k=j+1}^{n} \mathbf{A}_{k,j}^\top \boldsymbol{\varphi}_k \bar{\boldsymbol{\varphi}}_k^\top \mathbf{A}_{k,j} \\
&= \mathbf{M}_j + \mathbf{A}_{j+1,j}^\top \left(\widehat{\mathbf{M}}_{j+1} - \sum_{k=j+1}^{n} \mathbf{A}_{k,j+1}^\top \boldsymbol{\varphi}_k \bar{\boldsymbol{\varphi}}_k^\top \mathbf{A}_{k,j+1} \right) \mathbf{A}_{j+1,j} \\
&= \mathbf{M}_j + \mathbf{A}_{j+1,j}^\top \left(\widehat{\mathbf{M}}_{j+1} - \boldsymbol{\varphi}_{j+1} \bar{\boldsymbol{\varphi}}_{j+1}^\top - \sum_{k=j+2}^{n} \mathbf{A}_{k,j+1}^\top \boldsymbol{\varphi}_k \bar{\boldsymbol{\varphi}}_k^\top \mathbf{A}_{k,j+1} \right) \mathbf{A}_{j+1,j} \\
&= \mathbf{M}_j + \mathbf{A}_{j+1,j}^\top \left(\bar{\mathbf{M}}_{j+1} - \boldsymbol{\varphi}_{j+1} \bar{\boldsymbol{\varphi}}_{j+1}^\top \right) \mathbf{A}_{j+1,j}
\end{aligned}$$

using the definition of $\bar{\mathbf{M}}_{j+1}$. This last relation proves the recursion for $\bar{\mathbf{M}}$.
The remaining terms are the $\{\mathbf{U}_{i,j};\, \forall i = j-1 \dots 1\}$. The general identity (A.30) is now written for the off-diagonal terms (j,i) by:

$$\forall i = j-1 \dots 1, \quad \boldsymbol{\Delta}_j \mathbf{U}_{i,j}^\top + \sum_{k=j+1}^{n} \mathbf{U}_{j,k} \boldsymbol{\Delta}_k \mathbf{U}_{i,k}^\top = \mathbf{P}_j^\top \widehat{\mathbf{M}}_j \mathbf{A}_{j,i} \mathbf{P}_i$$

Resolving for $\mathbf{U}_{i,j}$ and applying the hypothesis of recursion for $\mathbf{U}_{i,k}$ with $k > j$, it yields:

$$\begin{aligned}
\mathbf{U}_{i,j} &= \left(\mathbf{P}_i^\top \mathbf{A}_{j,i}^\top \widehat{\mathbf{M}}_j \mathbf{P}_j - \sum_{k=j+1}^{n} \mathbf{U}_{i,k} \boldsymbol{\Delta}_k \mathbf{U}_{j,k}^\top \right) \boldsymbol{\Delta}_j^{-1} \\
&= \left(\mathbf{P}_i^\top \mathbf{A}_{j,i}^\top \widehat{\mathbf{M}}_j \mathbf{P}_j - \sum_{k=j+1}^{n} \mathbf{P}_i^\top \mathbf{A}_{k,i}^\top \bar{\mathbf{M}}_k \mathbf{P}_k \boldsymbol{\Delta}_k^{-1} \boldsymbol{\Delta}_k \boldsymbol{\Delta}_k^{-1} \mathbf{P}_k^\top \bar{\mathbf{M}}_k \mathbf{A}_{k,j} \mathbf{P}_j \right) \boldsymbol{\Delta}_j^{-1} \\
&= \mathbf{P}_i^\top \mathbf{A}_{j,i}^\top \left(\widehat{\mathbf{M}}_j - \sum_{k=j+1}^{n} \mathbf{A}_{k,j}^\top \boldsymbol{\varphi}_k \bar{\boldsymbol{\varphi}}_k^\top \mathbf{A}_{k,j} \right) \mathbf{P}_j \boldsymbol{\Delta}_j^{-1} \\
&= \mathbf{P}_i^\top \mathbf{A}_{j,i}^\top \bar{\mathbf{M}}_j \mathbf{P}_j \boldsymbol{\Delta}_j^{-1}
\end{aligned}$$

which was the last recursion to prove.

Since the three recursion are satisfied at step j, then the whole recursions in (A.33), (A.34) and (A.37) are demonstrated. □

With these analytical expressions of $\mathbf{U}$ and $\mathbf{\Delta}$, a parallel must be made with the initial decomposition of $\mathbf{D}$ through (A.19). The alternative decomposition proposed hereafter will be used in the proof of the forward dynamics algorithm for space robot.

Since the expression of $\mathbf{U}_{i,j}$ in (A.33) has a left product with $\mathbf{P}_i^\top$, the rectangular matrix $\tilde{\mathbf{U}}$ is introduced as follows using (A.21):

$$\mathbf{U} = \begin{bmatrix} \mathbf{I}_1 & \mathbf{P}_1^\top \tilde{\mathbf{U}}_{1,2} & \cdots & \mathbf{P}_1^\top \tilde{\mathbf{U}}_{1,n} \\ \mathbf{0} & \mathbf{I}_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{P}_n^\top \tilde{\mathbf{U}}_{n-1,n} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I}_1 \end{bmatrix} = \mathbf{N}_d^\top \underbrace{\begin{bmatrix} \tilde{\mathbf{U}}_{1,1} & \tilde{\mathbf{U}}_{1,2} & \cdots & \tilde{\mathbf{U}}_{1,n} \\ \mathbf{0} & \tilde{\mathbf{U}}_{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \tilde{\mathbf{U}}_{n-1,n} \\ \mathbf{0} & \cdots & \mathbf{0} & \tilde{\mathbf{U}}_{n,n} \end{bmatrix}}_{\triangleq \tilde{\mathbf{U}}} \tag{A.39}$$

where the terms of $\tilde{\mathbf{U}}$ are defined by $\tilde{\mathbf{U}}_{i,j} = \mathbf{A}_{j,i}^\top \boldsymbol{\varphi}_j$ according to the analytic expression of $\mathbf{U}_{i,j} = \mathbf{P}_i^\top \mathbf{A}_{j,i}^\top \boldsymbol{\varphi}_j$ in (A.33). In the same way, the identity blocks are expanded using the fact that $\mathbf{U}_{i,i} = \mathbf{P}_i^\top \boldsymbol{\varphi}_i = \mathbf{I}$, to define the diagonal terms by $\tilde{\mathbf{U}}_{i,i} = \boldsymbol{\varphi}_i$.

With this new matrix, the parallel with (A.19) leads to this fundamental equality for the demonstration of the space robot forward dynamics:

$$\boxed{\widehat{\mathbf{M}} = \mathbf{N}_l^\top \, [\mathbf{M}] \, \mathbf{N}_l = \tilde{\mathbf{U}} \, \mathbf{\Delta} \, \tilde{\mathbf{U}}^\top} \tag{A.40}$$

Let us come back to the initial task of inverting the global mass matrix to perform the forward dynamics. Knowing the analytical expression of the $\mathbf{U}\mathbf{\Delta}\mathbf{U}^\top$ decomposition, the system $\left(\mathbf{U}\mathbf{\Delta}\mathbf{U}^\top\right)\mathbf{X} = \mathbf{b}$ with $\mathbf{X}$ unknown, is solved in three steps:

1. Solve: $\mathbf{U}\widehat{\mathbf{X}} = \mathbf{b}$
2. Solve: $\mathbf{\Delta}\bar{\mathbf{X}} = \widehat{\mathbf{X}}$ $(= \mathbf{U}^{-1}\,\mathbf{b})$
3. Solve: $\mathbf{U}^\top\mathbf{X} = \bar{\mathbf{X}}$ $(= \mathbf{\Delta}^{-1}\,\mathbf{U}^{-1}\,\mathbf{b})$

Step 1: The system $\mathbf{U}\widehat{\mathbf{X}} = \mathbf{b}$ reads as follows:

$$\begin{bmatrix} \mathbf{I} & \mathbf{U}_{1,2} & \cdots & \mathbf{U}_{1,n} \\ \mathbf{0} & \mathbf{I} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{U}_{n-1,n} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{X}}_1 \\ \widehat{\mathbf{X}}_2 \\ \vdots \\ \widehat{\mathbf{X}}_n \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{bmatrix}$$

The algorithm is based on the recursive relation that links the i^{th} unknown $\widehat{\mathbf{X}}_i$ to the previous ones. The system is solved upward starting from the last row :

$$\begin{cases} \widehat{\mathbf{X}}_n = \mathbf{b}_n \\ \widehat{\mathbf{X}}_i = \mathbf{b}_i - \sum_{k=i+1}^{n} \mathbf{U}_{i,k}\widehat{\mathbf{X}}_k \qquad (\forall i = n-1 \dots 1) \end{cases}$$

By using the analytical expression of $\mathbf{U}_{i,j}$ obtained in (A.33), the following recursion is derived to solve the system efficiently based on the additional variable $\boldsymbol{\eta}_i$:

$$\begin{aligned} \widehat{\mathbf{X}}_i &= \mathbf{b} - \sum_{k=i+1}^{n} \mathbf{U}_{i,k}\widehat{\mathbf{X}}_k \\ &= \mathbf{b} - \sum_{k=i+1}^{n} \mathbf{P}_i^\top \mathbf{A}_{k,i}^\top \boldsymbol{\varphi}_k \widehat{\mathbf{X}}_k \\ &= \mathbf{b} - \mathbf{P}_i^\top \mathbf{A}_{i+1,i}^\top \underbrace{\left(\sum_{k=i+1}^{n} \mathbf{A}_{k,i+1}^\top \boldsymbol{\varphi}_k \widehat{\mathbf{X}}_k \right)}_{\triangleq \boldsymbol{\eta}_{i+1}} \end{aligned}$$

Developing further this term, a recursion is obtained as:

$$\begin{aligned} \boldsymbol{\eta}_{i+1} &= \boldsymbol{\varphi}_{i+1}\widehat{\mathbf{X}}_{i+1} + \sum_{k=i+2}^{n} \mathbf{A}_{k,i+1}^\top \boldsymbol{\varphi}_k \widehat{\mathbf{X}}_k \\ &= \boldsymbol{\varphi}_{i+1}\widehat{\mathbf{X}}_{i+1} + \mathbf{A}_{i+2,i+1}^\top \underbrace{\sum_{k=i+2}^{n} \mathbf{A}_{k,i+2}^\top \boldsymbol{\varphi}_k \widehat{\mathbf{X}}_k}_{=\boldsymbol{\eta}_{i+2}} \end{aligned}$$

This last relation is the foundation of the RGE technique. Starting with $\boldsymbol{\eta}_{n+1} = \mathbf{0}_{6\times 1}$, the

upper triangular system is solved as follows:

$$\forall i = n \dots 1, \quad \widehat{\mathbf{X}}_i = \mathbf{b}_i - \mathbf{P}_i^\top \mathbf{A}_{i+1,i}^\top \boldsymbol{\eta}_{i+1} \tag{A.41a}$$

$$\forall i = n \dots 1, \quad \boldsymbol{\eta}_i = \boldsymbol{\varphi}_i \widehat{\mathbf{X}}_i + \mathbf{A}_{i+1,i}^\top \boldsymbol{\eta}_{i+1} \tag{A.41b}$$

Step 2: The system $\boldsymbol{\Delta}\bar{\mathbf{X}} = \widehat{\mathbf{X}}$ is solved for $\bar{\mathbf{X}}$ provided that **Step 1** yields $\widehat{\mathbf{X}}$:

$$\begin{bmatrix} \boldsymbol{\Delta}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Delta}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\Delta}_n \end{bmatrix} \begin{bmatrix} \bar{\mathbf{X}}_1 \\ \bar{\mathbf{X}}_2 \\ \vdots \\ \bar{\mathbf{X}}_n \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{X}}_1 \\ \widehat{\mathbf{X}}_2 \\ \vdots \\ \widehat{\mathbf{X}}_n \end{bmatrix}$$

The solution is straightforward, and is given by:

$$\forall i = 1 \dots n, \quad \bar{\mathbf{X}}_i = \boldsymbol{\Delta}_i^{-1} \widehat{\mathbf{X}}_i \tag{A.42}$$

Step 3: Eventually, the system $\mathbf{U}^\top \mathbf{X} = \bar{\mathbf{X}}$ is solved using the same modified RGE algorithm on:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{U}_{1,2}^\top & \mathbf{I} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{U}_{1,n}^\top & \cdots & \mathbf{U}_{n-1,n}^\top & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_n \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{X}}_1 \\ \bar{\mathbf{X}}_2 \\ \vdots \\ \bar{\mathbf{X}}_n \end{bmatrix}$$

To solve this lower triangular system, the recursion goes downward by starting from the first row :

$$\begin{cases} \mathbf{X}_1 = \bar{\mathbf{X}}_1 \\ \mathbf{X}_i = \bar{\mathbf{X}}_i - \sum_{k=1}^{i-1} \mathbf{U}_{k,i}^\top \mathbf{X}_k \qquad (\forall i = 2 \dots n) \end{cases}$$

Using the same kind of computation, the lower triangular system is solved with (A.33) as

follows:

$$\begin{aligned}\mathbf{X}_i &= \bar{\mathbf{X}}_i - \sum_{k=1}^{i-1} \mathbf{U}_{k,i}^{\top} \mathbf{X}_k \\ &= \bar{\mathbf{X}}_i - \sum_{k=1}^{i-1} \boldsymbol{\varphi}_i^{\top} \mathbf{A}_{i,k} \mathbf{P}_k \mathbf{X}_k \\ &= \bar{\mathbf{X}}_i - \boldsymbol{\varphi}_i^{\top} \mathbf{A}_{i,i-1} \underbrace{\left(\sum_{k=1}^{i-1} \mathbf{A}_{i-1,k} \mathbf{P}_k \mathbf{X}_k \right)}_{\triangleq \boldsymbol{\mu}_{i-1}}\end{aligned}$$

Again, the expression of $\boldsymbol{\mu}_{i-1}$ is developed to obtain a recursive relation with its predecessor:

$$\begin{aligned}\boldsymbol{\mu}_{i-1} &= \sum_{k=1}^{i-1} \mathbf{A}_{i-1,k} \mathbf{P}_k \mathbf{X}_k \\ &= \mathbf{P}_{i-1} \mathbf{X}_{i-1} + \sum_{k=1}^{i-2} \mathbf{A}_{i-1,k} \mathbf{P}_k \mathbf{X}_k \\ &= \mathbf{P}_{i-1} \mathbf{X}_{i-1} + \mathbf{A}_{i-1,i-2} \underbrace{\sum_{k=1}^{i-2} \mathbf{A}_{i-2,k} \mathbf{P}_k \mathbf{X}_k}_{=\mu_{i-2}}\end{aligned}$$

Similar to (A.41), but starting with $\boldsymbol{\mu}_0 = \mathbf{0}_{6\times 1}$:

$$\forall i = 1 \ldots n, \quad \mathbf{X}_i = \bar{\mathbf{X}}_i - \boldsymbol{\varphi}_i^{\top} \mathbf{A}_{i,i-1} \boldsymbol{\mu}_{i-1} \tag{A.43a}$$

$$\forall i = 1 \ldots n, \quad \boldsymbol{\mu}_i = \mathbf{P}_i \, \mathbf{X}_i \, + \, \mathbf{A}_{i,i-1} \, \boldsymbol{\mu}_{i-1} \tag{A.43b}$$

The whole forward dynamics algorithm is given in Algorithm A.3 based on the three recursions detailed in (A.41), (A.42) and (A.43).

Algorithm A.3: Forward Dynamics Algorithm based on DeNOC approach

Function : $\ddot{\mathbf{q}} = \texttt{ForDyn}(\mathbf{b})$
Input : $\mathbf{b}$
Output : $\ddot{\mathbf{q}}$
Data : $\{\mathbf{A}_{i,i-1}\,,\,\mathbf{P}_i\,,\,\mathbf{M}_i\,;\; i=1\ldots n+1\}$

Computation and storage of $\mathbf{A}_{i,i-1}$ // **KINETICS LOOP**
(usually done in previous computation of $\mathbf{h}_\mathrm{r}$ or $\mathbf{C}_\mathrm{r}$)

`Initialize` $\left(\boldsymbol{\eta}_{n+1}=\mathbf{0}_{6\times1},\,\bar{\mathbf{M}}_{n+1}=\mathbf{M}_{n+1},\,\bar{\boldsymbol{\varphi}}_{n+1}=\boldsymbol{\varphi}_{n+1}=\mathbf{0}_{6\times1}\right)$ // **UPPER TRIANGULAR SYSTEM**

for $i=n\ldots1$ **do**

$\boldsymbol{\eta}_{i,i+1}=\mathbf{A}_{i+1,i}^{\top}\,\boldsymbol{\eta}_{i+1}$ // Recursive computation of $\mathbf{U}\hat{\mathbf{X}}=\mathbf{b}$

$\hat{\mathbf{X}}_i=\mathbf{b}_i-\mathbf{P}_i^{\top}\,\boldsymbol{\eta}_{i,i+1}$ (A.41a)

$\bar{\mathbf{M}}_i=\mathbf{M}_i+\mathbf{A}_{i+1,i}^{\top}\left(\bar{\mathbf{M}}_{i+1}-\boldsymbol{\varphi}_{i+1}\,\bar{\boldsymbol{\varphi}}_{i+1}^{\top}\right)\mathbf{A}_{i+1,i}$ (A.34)

$\bar{\boldsymbol{\varphi}}_i=\bar{\mathbf{M}}_i\,\mathbf{P}_i$ (A.35)

$\boldsymbol{\Delta}_i=\mathbf{P}_i^{\top}\,\bar{\boldsymbol{\varphi}}_i$ (A.37)

$\boldsymbol{\varphi}_i=\bar{\boldsymbol{\varphi}}_i\boldsymbol{\Delta}_i^{-1}$ (A.36)

$\boldsymbol{\eta}_i=\boldsymbol{\varphi}_i\hat{\mathbf{X}}_i\,+\,\boldsymbol{\eta}_{i,i+1}$ (A.41b)

end

`Initialize` $\left(\emptyset\right)$ // **DIAGONAL SYSTEM**

for $i=1\ldots n$ **do**

$\bar{\mathbf{X}}_i=\boldsymbol{\Delta}_i^{-1}\,\hat{\mathbf{X}}_i$ (A.42)

end

`Initialize` $\left(\boldsymbol{\mu}_0=\mathbf{0}_{6\times1}\right)$ // **LOWER TRIANGULAR SYSTEM**

for $i=1\ldots n$ **do**

$\boldsymbol{\mu}_{i,i-1}=\mathbf{A}_{i,i-1}\boldsymbol{\mu}_{i-1}$ // Recursive computation of $\mathbf{U}^{\top}\mathbf{X}=\bar{\mathbf{X}}$

$\mathbf{X}_i=\bar{\mathbf{X}}_i-\boldsymbol{\varphi}_i^{\top}\,\boldsymbol{\mu}_{i,i-1}$ (A.43a)

$\boldsymbol{\mu}_i=\mathbf{P}_i\,\mathbf{X}_i\,+\,\boldsymbol{\mu}_{i,i-1}$ (A.43b)

end

---

return $\ddot{\mathbf{q}}=\mathbf{X}$

APPENDIX B FLEXIBLE MODELING DETAILS

This appendix gathers all the results and computations used in the modeling of flexible manipulators.

The first section is dedicated to the computation of the flexible modes using the *Assumed Mode Method (AMM)*. The theory is quickly covered to focus on the clamped-loaded boundary conditions used to describe the flexible segments of a manipulator. The mode shapes are stored in compact matrices to describe flexibility in each direction of space through traction, bending and torsion.

In the second part, these results are used to compute the flexible kinetic energy through the rigid/flexible decomposition of speed. Each term is developed separately to simplify its derivation, and the flexible mass matrix is introduced to express the global kinetic energy. This analytical expression of the mass matrix is the foundation of the flexible modeling, by its decomposition into the nominal terms and the ones depending explicitly on the flexible coordinates.

Finally, the last section deals with the derivation of the quasi-Lagrangian equations of motion for the previous kinetic energy. The time and state derivatives are performed to provide the dynamic model under matrix form, free of any integral over space, in order to provide less on-line computations during the simulation. This model will allow to easily highlight the terms that can be neglected during the design of approximate models in simulation.

B.1 Assumed Mode Method

B.1.1 Assumed Mode Formalism

The *Assumed Mode Method (AMM)* is used to model the four different directions of flexibility presented in Figure B.1 to recall Figure 5.2. The slender part of the segment is supposed to behave like a beam.

Traction and torsion deformations obey the same equation, called the *wave equation*. For a constant section area S and density ρ, a polar second moment of area I_p, a Young modulus

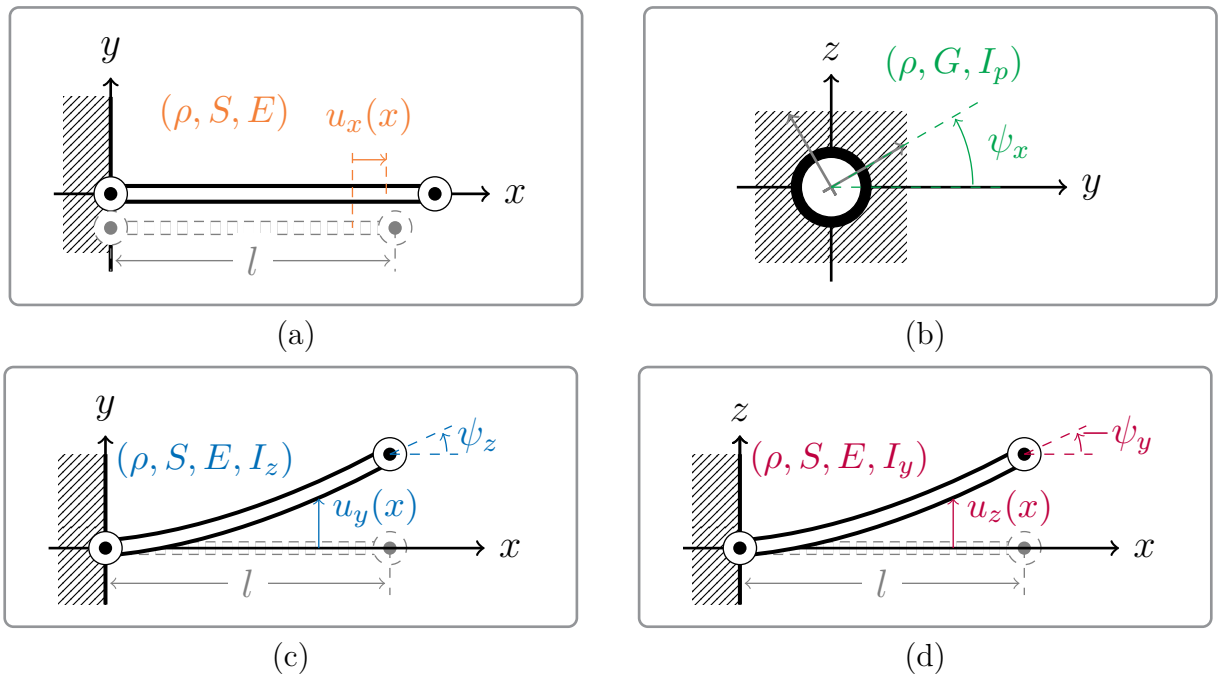


Figure B.1: Flexible behavior of the beam in each direction; (a) Traction along X axis; (b) Torsion around X axis; (c) Bending in (X, Y) plane; (d) Bending in (X, Z) plane.

E and a Poisson modulus G, flexible deformations u_x and ψ_x satisfy the following PDEs (Meirovitch, 2001):

$$ES\frac{\partial^2 u_x(s,t)}{\partial s^2} - \rho S\frac{\partial^2 u_x(s,t)}{\partial t^2} = 0 \tag{B.1}$$

$$GI_p\frac{\partial^2 \psi_x(s,t)}{\partial s^2} - \rho I_p\frac{\partial^2 \psi_x(x,t)}{\partial t^2} = 0 \tag{B.2}$$

The bending modes computation is based on the Euler-Bernoulli theory. For a constant second moment of area I_z (resp. I_y for z-bending in (X, Z) plane), bending deformation u_y (resp. u_z) obeys the following PDE:

$$EI_{z/y}\frac{\partial^4 u_{y/z}(s,t)}{\partial s^4} + \rho S\frac{\partial^2 u_{y/z}(s,t)}{\partial t^2} = 0 \tag{B.3}$$

With the AMM, each deformation is written as the summation of several flexible modes. Assuming the separation in time and space, each of them is the product of a mode shape $\phi_i(x)$ by a time-varying amplitude $\delta_i(t)$. In the general case, a flexible deformation reads:

$$u_f(x,t) = \sum_{i=1}^{\infty} \phi_i(x)\,\delta_i(t)$$

When inserting the time-varying amplitudes under harmonic form[1], the previous PDEs yield ODEs which are only function of the curvilinear abscissa s. The general solution for each flexible direction is given by:

$$\begin{aligned}
\phi_x(s) &= A_x\ \cos(k_x s) + B_x\ \sin(k_x s) \\
\phi_\alpha(s) &= A_\alpha\ \cos(k_\alpha s) + B_\alpha\ \sin(k_\alpha s) \\
\phi_y(s) &= A_y\ \cos(k_y s) + B_y\ \sin(k_y s) + C_y\ \cosh(k_y s) + D_y\ \sinh(k_y s) \\
\phi_z(s) &= A_z\ \cos(k_z s) + B_z\ \sin(k_z s) + C_z\ \cosh(k_z s) + D_z\ \sinh(k_z s)
\end{aligned}$$

where $k_x^2 = \frac{\rho\,\omega_x^2}{E}$, $k_\alpha^2 = \frac{\rho\,\omega_\alpha^2}{G}$ and $k_{y/z}^4 = \frac{\rho S\,\omega_{y/z}^2}{E I_{z/y}}$ are the spatial pulsations corresponding to the time pulsations $\omega_{x/\alpha/y/z}$ in, respectively, traction, torsion and bending. These time and spatial pulsations are solutions of transcendental equations, which must be solved numerically and depends on the load mass and inertia. They are detailed in the following.

B.1.2 Boundary Conditions

Mode coefficients (A_i, B_i, C_i, D_i) are obtained by expressing the boundary conditions satisfied at both segment tips. In the scope of a multi-link manipulator, the *clamped-loaded* boundary conditions are considered. The clamped assumption is made because the model is derived in the local frame, in which the segment base is always fixed. Furthermore, the loaded assumption at segment's end-tip is due to inertia and mass of the next segments. For a beam of length l, the boundary conditions on u_x, ψ_x, u_y and u_z are summarized for a load of mass M_L and inertia tensor J_L in Tab. B.1.

These relations yield a system of linear equations of the form :

$$\left[\ \mathbf{F}(k_{x/\alpha})\ \right]_{2\times 2} \begin{bmatrix} A_{x/\alpha} \\ B_{x/\alpha} \end{bmatrix} = 0 \quad \text{and} \quad \left[\ \mathbf{G}(k_{y/z})\ \right]_{4\times 4} \begin{bmatrix} A_{y/z} \\ B_{y/z} \\ C_{y/z} \\ D_{y/z} \end{bmatrix} = 0 \tag{B.4}$$

which admits a non-zero solution if and only if the matrices $\mathbf{F}$ and $\mathbf{G}$ are singular (i.e. if $\det(\mathbf{F}(k_{x/\alpha})) = 0$ and $\det(\mathbf{G}(k_{y/z})) = 0$). These last relations are transcendental in spatial pulsation $k_{x/\alpha/y/z}$ and can only be solved numerically for a given load mass and inertia. After some algebraic manipulations, one gets the following equations to obtain the mode pulsations

[1] i.e., $\phi_i(t) = e^{j\omega_i t}$ with ω_i the i^{th} mode pulsation

Table B.1: Boundary conditions for a clamped-loaded beam

	Base conditions	**End-tip conditions**
Traction T_x	$u_x(0,t) = 0$	$ES\,\frac{\partial u_x}{\partial s}(l,t) = -M_L\,\frac{d^2}{dt^2}\,(u_x(l,t))$
Torsion R_x	$\psi_x(0,t) = 0$	$\rho I_p\,\frac{\partial \psi_x}{\partial s}(l,t) = -J_L^x\,\frac{d^2}{dt^2}\,(\psi_x(l,t))$
Y**-bending** (T_y, R_z)	$u_y(0,t) = 0$	$EI_z\,\frac{\partial^2 u_y}{\partial s^2}(l,t) = -J_L^z\,\frac{d^2}{dt^2}\left(\frac{\partial u_y}{\partial s}(l,t)\right)$
	$\frac{\partial u_y}{\partial s}(0,t) = 0$	$EI_z\,\frac{\partial^3 u_y}{\partial s^3}(l,t) = +M_L\,\frac{d^2}{dt^2}\,(u_y(l,t))$
Z**-bending** (T_z, R_y)	$u_z(0,t) = 0$	$EI_y\,\frac{\partial^2 u_z}{\partial s^2}(l,t) = -J_L^y\,\frac{d^2}{dt^2}\left(\frac{\partial u_z}{\partial s}(l,t)\right)$
	$\frac{\partial u_z}{\partial s}(0,t) = 0$	$EI_y\,\frac{\partial^3 u_z}{\partial s^3}(l,t) = +M_L\,\frac{d^2}{dt^2}\,(u_z(l,t))$

in traction and torsion:

$$\cos(k_x l) - \frac{M_L}{\rho S} k_x\,\sin(k_x l) \;=\; 0 \tag{B.5}$$

$$\cos(k_\alpha l) - \frac{J_L^x}{\rho I_p} k_\alpha\,\sin(k_\alpha l) \;=\; 0 \tag{B.6}$$

while bending pulsations are given by:

$$\begin{aligned} &\left(-1-\cos(k_{y/z}l)\,\cosh(k_{y/z}l)\right) \\ &+\tfrac{M_L J_L^{z/y}}{(\rho S)^2}k_{y/z}^4\left(-1+\cos(k_{y/z}l)\,\cosh(k_{y/z}l)\right) \\ &+\tfrac{J_L^{z/y}}{\rho S}k_{y/z}^3\left(\sin(k_{y/z}l)\,\cosh(k_{y/z}l)+\cos(k_{y/z}l)\,\sinh(k_{y/z}l)\right) \\ &+\tfrac{M_L}{\rho S}k_{y/z}\left(\sin(k_{y/z}l)\,\cosh(k_{y/z}l)-\cos(k_{y/z}l)\,\sinh(k_{y/z}l)\right) \;=\; 0 \end{aligned} \tag{B.7}$$

The mode pulsations $\omega_{x/\alpha/y/z}$ are found by solving numerically these equations for a given load, given by (M_L, J_L). But according to (B.4), the mode coefficients are defined up to a multiplicative constant since $\mathbf{F}$ and $\mathbf{G}$ are singular. This last degree-of-freedom is set by normalizing the mode integrals. The normalization value is usually chosen to simplify the flexible dynamics equations. In (De Luca and Siciliano, 1991) for example, it is advised to

set :

$$\int_0^l \rho S\, \phi_{x/\alpha/y/z}^2 ds = 1 \quad \text{or} \quad \left\{ \begin{array}{rcl} \int_0^l ES\, \phi_x^{'\,2} ds & = & \omega_x^2 \\ \int_0^l GI_p\, \phi_\alpha^{'\,2} ds & = & \omega_\alpha^2 \\ \int_0^l EI_{z,y}\, \phi_{y/z}^{''\,2} ds & = & \omega_{y/z}^2 \end{array} \right. \tag{B.8}$$

B.1.3 Flexible Displacements

For a numerical implementation, a finite number of modes is considered and the flexible deformations are given by finite sums, as follows:

$$u_f(x,t) = \sum_{i=1}^{n_f} \phi_i(x)\, \delta_i(t) \triangleq \boldsymbol{\phi}^\top(x)\, \boldsymbol{\delta}(t)$$

For a truncation of n_x modes for traction, n_y and n_z for bending, and n_α in torsion, the total number of modes is $n_f = n_x + n_y + n_z + n_\alpha$. Traction mode shapes and amplitudes are denoted respectively $\boldsymbol{\phi}_x$ and $\boldsymbol{\delta}_x$, while y, z and α subscripts are used for bending in both y and z-direction, and for torsion.

With these notations, the linear deformation of a point of coordinates (y,z) in the cross-section plane located at the curvilinear abscissa s is given by (Damaren and Sharf, 1995):

$$\mathbf{u} = \begin{bmatrix} \boldsymbol{\phi}_x^\top(s) \\ \mathbf{0}_{1\times n_x} \\ \mathbf{0}_{1\times n_x} \end{bmatrix} \boldsymbol{\delta}_x(t) + \begin{bmatrix} -y\, \boldsymbol{\phi}_y'^\top(s) \\ \boldsymbol{\phi}_y^\top(s) \\ \mathbf{0}_{1\times n_y} \end{bmatrix} \boldsymbol{\delta}_y(t) + \begin{bmatrix} -z\, \boldsymbol{\phi}_z'^\top(s) \\ \mathbf{0}_{1\times n_z} \\ \boldsymbol{\phi}_z^\top(s) \end{bmatrix} \boldsymbol{\delta}_z(t) + \begin{bmatrix} \mathbf{0}_{1\times n_\alpha} \\ -z\, \boldsymbol{\phi}_\alpha^\top(s) \\ y\, \boldsymbol{\phi}_\alpha^\top(s) \end{bmatrix} \boldsymbol{\delta}_\alpha(t) \tag{B.9}$$

where $\boldsymbol{\delta}_{x/y/z/\alpha}$ and $\boldsymbol{\delta}_{x/y/z/\alpha} \in \mathbb{R}^{n_{x/y/z/\alpha}}$. Superscript $\{.\}'$, as $\boldsymbol{\phi}_y'$, stands for spatial derivative, used to approximate the angular deflections.

The terms depending on the transverse position (y,z) are actually due to the flexible rotation of the cross-section. They are based on the following discretization of the flexible rotation of the cross-section:

$$\boldsymbol{\psi} = \begin{bmatrix} \mathbf{0}_{1\times n_y} \\ \mathbf{0}_{1\times n_y} \\ \boldsymbol{\phi}_y'^\top(s) \end{bmatrix} \boldsymbol{\delta}_y(t) + \begin{bmatrix} \mathbf{0}_{1\times n_z} \\ -\boldsymbol{\phi}_z'^\top(s) \\ \mathbf{0}_{1\times n_z} \end{bmatrix} \boldsymbol{\delta}_z(t) + \begin{bmatrix} \boldsymbol{\phi}_\alpha^\top(s) \\ \mathbf{0}_{1\times n_\alpha} \\ \mathbf{0}_{1\times n_\alpha} \end{bmatrix} \boldsymbol{\delta}_\alpha(t) \tag{B.10}$$

Indeed, this last relation is based on the small angles hypothesis with regards to the flexible rotations ($\psi_x, \psi_y, \psi_z \ll 1$), such that $\psi_z \simeq \frac{\partial u_y}{\partial s}$ and $\psi_y \simeq -\frac{\partial u_z}{\partial s}$ as shown in Figure B.1.

The corresponding flexible variables are merged into the vector :

$$\boldsymbol{\delta}(t) = \left[\begin{array}{cccc} \boldsymbol{\delta}_x^\top(t) & \boldsymbol{\delta}_y^\top(t) & \boldsymbol{\delta}_z^\top(t) & \boldsymbol{\delta}_\alpha^\top(t) \end{array} \right]^\top \tag{B.11}$$

while the translational and rotational modes are stored in :

$$\boldsymbol{\phi}_u(s) = \left[\begin{array}{cccc} \boldsymbol{\phi}_x^\top(s) & \boldsymbol{\phi}_y^\top(s) & \boldsymbol{\phi}_z^\top(s) & \mathbf{0}_{n_\alpha \times 1}^\top \end{array} \right]^\top \tag{B.12}$$

$$\boldsymbol{\phi}_\psi(s) = \left[\begin{array}{cccc} \mathbf{0}_{n_x \times 1}^\top & \boldsymbol{\phi}_y'^\top(s) & -\boldsymbol{\phi}_z'^\top(s) & \boldsymbol{\phi}_\alpha^\top(s) \end{array} \right]^\top \tag{B.13}$$

In order to highlight the influence of the beam section rotation, the total flexible displacement $\mathbf{u}$ is split between the centroidal axis motion $\mathbf{u}_c$ (i.e., for $(y,z) = (0,0)$), and the transverse one $\delta\mathbf{u}$ in (5.6). The equations (B.9) and (B.10) are thus written for the centroidal axis first:

$$\mathbf{u}_c(s,t) = \underbrace{\left[\begin{array}{ccc|c} \boldsymbol{\phi}_x^\top(s) & \mathbf{0}_{1\times n_y} & \mathbf{0}_{1\times n_z} & \mathbf{0}_{1\times n_\alpha} \\ \mathbf{0}_{1\times n_x} & \boldsymbol{\phi}_y^\top(s) & \mathbf{0}_{1\times n_z} & \mathbf{0}_{1\times n_\alpha} \\ \mathbf{0}_{1\times n_x} & \mathbf{0}_{1\times n_y} & \boldsymbol{\phi}_z^\top(s) & \mathbf{0}_{1\times n_\alpha} \end{array} \right]}_{\triangleq \boldsymbol{\Phi}(s)} \boldsymbol{\delta}(t) \tag{B.14a}$$

and

$$\boldsymbol{\psi}(s,t) = \underbrace{\left[\begin{array}{ccc|c} \mathbf{0}_{1\times n_x} & \mathbf{0}_{1\times n_y} & \mathbf{0}_{1\times n_z} & \boldsymbol{\phi}_\alpha^\top(s) \\ \mathbf{0}_{1\times n_x} & \mathbf{0}_{1\times n_y} & -\boldsymbol{\phi}_z'^\top(s) & \mathbf{0}_{1\times n_\alpha} \\ \mathbf{0}_{1\times n_x} & \boldsymbol{\phi}_y'^\top(s) & \mathbf{0}_{1\times n_z} & \mathbf{0}_{1\times n_\alpha} \end{array} \right]}_{\triangleq \boldsymbol{\Delta}(s)} \boldsymbol{\delta}(t) \tag{B.14b}$$

while the flexible displacement due to transverse terms is given by:

$$\delta\mathbf{u}(\mathbf{p},t) = \left[\begin{array}{ccc|c} \mathbf{0}_{1\times n_x} & -y\,\boldsymbol{\phi}_y'^\top(s) & -z\,\boldsymbol{\phi}_z'^\top(s) & \mathbf{0}_{1\times n_\alpha} \\ \mathbf{0}_{1\times n_x} & \mathbf{0}_{1\times n_y} & \mathbf{0}_{1\times n_z} & -z\,\boldsymbol{\phi}_\alpha^\top(s) \\ \mathbf{0}_{1\times n_x} & \mathbf{0}_{1\times n_y} & \mathbf{0}_{1\times n_z} & y\,\boldsymbol{\phi}_\alpha^\top(s) \end{array} \right]$$

A careful inspection of this matrix allows to rewrite it in a more compact way:

$$\delta\mathbf{u}(\mathbf{p},t) = \boldsymbol{\psi}(s,t) \times \left[\begin{array}{c} 0 \\ y \\ z \end{array} \right] = \left(\boldsymbol{\Delta}(s)\,\boldsymbol{\delta}(t)\right)^\times \left[\begin{array}{c} 0 \\ y \\ z \end{array} \right] \tag{B.15}$$

using the Gibbs notation "$\times$" to denote the cross product.

B.2 Computation of the Flexible Mass Matrix

The computation of the flexible mass matrix is derived in this section. Using the kinetic energy partitioning into rigid and flexible terms in (5.15), recalled in (B.17), each term is developed explicitly based on the corresponding linear velocities in (5.14), recalled in (B.16). In the following, $\mathbf{I}$, $\mathbf{P}$ and $\mathbf{H}$ are used to denote the flexible contributions to, respectively, the second moment of inertia, and to the linear and angular momentums.

Reminder:
It is recalled that the augmented twist at the segment base is defined by merging the linear and angular velocity of the segment reference point O, with the flexible variables used to describe its deformation. It reads as follows in (5.9) (Mohan and Saha, 2009):

$$\mathbf{t}_{\mathrm{o}} = \begin{bmatrix} \boldsymbol{v}_{\mathrm{o}}^{\top} & \boldsymbol{\omega}_{\mathrm{o}}^{\top} & \dot{\boldsymbol{\delta}}^{\top} \end{bmatrix}^{\top}$$

Using this twist, the linear velocity of any point P along the segment is described with respect to this reference point, and to the flexible deformations in Appendix B.1:

$$\mathbf{v}_r = \begin{bmatrix} \mathbf{I}_3 & -\mathbf{p}^{\times} & \mathbf{0}_{3\times n_f} \end{bmatrix} \mathbf{t}_{\mathrm{o}} \tag{B.16a}$$

$$\mathbf{v}_u = \begin{bmatrix} \mathbf{0}_{3\times 3} & -(\boldsymbol{\Phi}\,\boldsymbol{\delta})^{\times} & \boldsymbol{\Phi} \end{bmatrix} \mathbf{t}_{\mathrm{o}} \tag{B.16b}$$

$$\mathbf{v}_\psi = \begin{bmatrix} \mathbf{0}_{3\times 3} & (\delta\mathbf{p}^{\times}\,(\boldsymbol{\Delta}\,\boldsymbol{\delta}))^{\times} & -\delta\mathbf{p}^{\times}\boldsymbol{\Delta} \end{bmatrix} \mathbf{t}_{\mathrm{o}} \tag{B.16c}$$

such that the kinetic energy has been split up according to:

$$\begin{aligned} T = {} & \underbrace{\frac{1}{2}\int_{\mathcal{L}} \mathbf{v}_r^{\top}\mathbf{v}_r \mathrm{d}m}_{T_{\mathrm{rr}}} + \underbrace{\frac{1}{2}\int_{\mathcal{B}} \mathbf{v}_u^{\top}\mathbf{v}_u \mathrm{d}m}_{T_{\mathrm{uu}}} + \underbrace{\frac{1}{2}\int_{\mathcal{B}} \mathbf{v}_\psi^{\top}\mathbf{v}_\psi \mathrm{d}m}_{T_{\psi\psi}} \\ & + \underbrace{\int_{\mathcal{B}} \mathbf{v}_r^{\top}\mathbf{v}_u \mathrm{d}m}_{T_{\mathrm{ru}}+T_{\mathrm{ur}}} + \underbrace{\int_{\mathcal{B}} \mathbf{v}_r^{\top}\mathbf{v}_\psi \mathrm{d}m}_{T_{\mathrm{r}\psi}+T_{\psi\mathrm{r}}} + \underbrace{\int_{\mathcal{B}} \mathbf{v}_u^{\top}\mathbf{v}_\psi \mathrm{d}m}_{T_{\mathrm{u}\psi}+T_{\psi\mathrm{u}}} \end{aligned} \tag{B.17}$$

B.2.1 Rigid Contribution

T_{rr} is developed using the term $\mathbf{v}_r$ in (B.16a). It yields the classic mass matrix of a rigid body, already obtained for the rigid case in (4.8) (Alazard et al., 2008):

$$\begin{aligned} T_{\mathrm{rr}} &= \frac{1}{2}\int_{\mathcal{L}} \mathbf{v}_r^{\top}\,\mathbf{v}_r\,\mathrm{d}m \\ &= \frac{1}{2}\int_{\mathcal{L}} \mathbf{t}_{\mathrm{o}}^{\top} \left[\begin{array}{cc|c} \mathbf{I}_3 & -\mathbf{p}^{\times} & \mathbf{0}_{3\times n_f} \\ * & -(\mathbf{p}^{\times})^2 & \mathbf{0}_{3\times n_f} \\ \hline * & * & \mathbf{0}_{n_f\times n_f} \end{array}\right] \mathbf{t}_{\mathrm{o}}\,\mathrm{d}m \end{aligned}$$

summarized by:

$$T_{\text{rr}} = \frac{1}{2}\,\mathbf{t}_{\text{o}}^{\top} \left[\begin{array}{cc|c} m\mathbf{I}_3 & -\mathbf{c}^{\times} & \mathbf{0}_{3\times n_f} \\ * & \mathbf{I}_{\text{o}} & \mathbf{0}_{3\times n_f} \\ \hline * & * & \mathbf{0}_{n_f\times n_f} \end{array}\right] \mathbf{t}_{\text{o}} \tag{B.18}$$

where it is recalled that m denotes the segment's mass; $\mathbf{c} = \int_{\mathcal{L}} \mathbf{p}\mathrm{d}m$ is the *first moment of inertia* defined by $\mathbf{c}/m = \mathbf{p}_G$, with $\mathbf{p}_G$ the relative position of its CoM w.r.t. O in $\mathcal{R}_{\text{o}}$; and $\mathbf{I}_{\text{o}} = \int_{\mathcal{L}}(-(\mathbf{p}^{\times})^2)\mathrm{d}m$ is the *second moment of inertia* of the undeformed body, expressed at O in $\mathcal{R}_{\text{o}}$ (Hughes, 1986; Spong et al., 2006). The symbol "$*$" represents the symmetric parts of a matrix.

B.2.2 Flexible Contributions

To express the flexible terms, the undeformed kinematics illustrated in Fig. B.2 is split according to the rigid part at the segment beginning, described by the vector $\mathbf{p}_{\text{a}}$, and to its slender part, along the unitary vector $\mathbf{e}_x$.

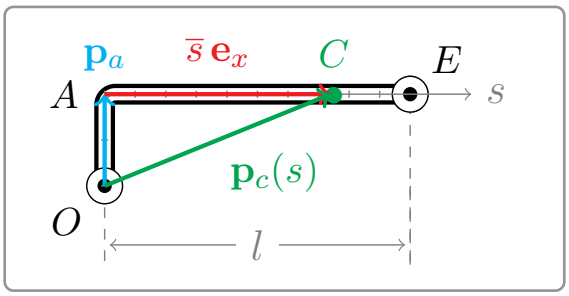


Figure B.2: Undeformed kinematics along the segment slender part

With these notations, the position along the centroidal axis can be parametrized on the beam section $\mathcal{B}$, as follows:

$$\mathbf{p}_c(s) = \mathbf{p}_a + \overline{s}\,\mathbf{e}_x \tag{B.19}$$

where a reduced curvilinear abscissa $\overline{s}$ is introduce along the flexible part, and goes from 0 to l, with l the segment's length.

B.2.2.1 Preliminary definitions

Before deriving the kinetic terms, the following matrices are introduced to denote the integrals of mode shapes. They are supposed to be computed off-line to alleviate the computational burden during the simulation (De Luca and Siciliano, 1991). Two mains categories of terms arise: the *translational* ones and the *rotary* ones (Cyril, 1988). Translational modes in $\boldsymbol{\phi}_u$ are due to the beam deformation in traction and bending, while rotary modes in $\boldsymbol{\phi}_\psi$ are caused by the cross-section rotation with torsion and bending.

The following matrices are introduced to represent the flexible modes integrals:

$$\mathbf{V}_u = \int_0^l \rho S\, \mathbf{\Phi}(\overline{s})\, \mathrm{d}\overline{s} \tag{B.20}$$

$$\mathbf{W}_u = \int_0^l \rho S\, \overline{s}\, \mathbf{\Phi}(\overline{s})\, \mathrm{d}\overline{s} \tag{B.21}$$

$$\mathbf{V}_\psi = \int_0^l \rho\, \mathbf{\Delta}(\overline{s})\, \mathrm{d}\overline{s} \tag{B.22}$$

where their subscripts are chosen in agreement with their translational or rotary nature. $\mathbf{V}_u \in \mathbb{R}^{3\times n_f}$ represents the flexible contribution to the linear momentum, while $\mathbf{V}_\psi \in \mathbb{R}^{3\times n_f}$ stands for the contribution to the angular momentum. The last term $\mathbf{W}_u \in \mathbb{R}^{3\times n_f}$ is a translational corrective term on the angular momentum.

Two matrices are also introduced to denote the *translational* and *rotary* inertias, by, respectively, $\mathbf{Z}_{uu}$ and $\mathbf{Z}_{\psi\psi}$. They represent the cross-products between the corresponding sets of flexible modes, $\boldsymbol{\phi}_u$ in (B.12) and $\boldsymbol{\phi}_\psi$ in (B.13) of Appendix B.1.

Firstly, the full matrix $\mathbf{Z}_{uu} \in \mathbb{R}^{n_f\times n_f}$ represents the cross-products between the translational modes in all directions:

$$\mathbf{Z}_{uu} \triangleq \int_0^l \rho S\, \boldsymbol{\phi}_u(\overline{s})\, \boldsymbol{\phi}_u^\top(\overline{s})\mathrm{d}\overline{s} = \int_0^l \rho S \left[\begin{array}{ccc|c} \boldsymbol{\phi}_x\boldsymbol{\phi}_x^\top & \boldsymbol{\phi}_x\boldsymbol{\phi}_y^\top & \boldsymbol{\phi}_x\boldsymbol{\phi}_z^\top & \mathbf{0}_{n_x\times n_\alpha} \\ * & \boldsymbol{\phi}_y\boldsymbol{\phi}_y^\top & \boldsymbol{\phi}_y\boldsymbol{\phi}_z^\top & \mathbf{0}_{n_y\times n_\alpha} \\ * & * & \boldsymbol{\phi}_z\boldsymbol{\phi}_z^\top & \mathbf{0}_{n_z\times n_\alpha} \\ \hline * & * & * & \mathbf{0}_{n_\alpha\times n_\alpha} \end{array}\right] \mathrm{d}\overline{s}$$

$$= \left[\begin{array}{ccc|c} \mathbf{Z}_{xx} & \mathbf{Z}_{xy} & \mathbf{Z}_{xz} & \mathbf{0}_{n_x\times n_\alpha} \\ * & \mathbf{Z}_{yy} & \mathbf{Z}_{yz} & \mathbf{0}_{n_y\times n_\alpha} \\ * & * & \mathbf{Z}_{zz} & \mathbf{0}_{n_z\times n_\alpha} \\ \hline * & * & * & \mathbf{0}_{n_\alpha\times n_\alpha} \end{array}\right]$$

the diagonal version of which stands for these cross-products in each direction separately, with:

$$\bar{\mathbf{Z}}_{\boldsymbol{uu}} \triangleq \int_0^l \rho S\, \mathbf{\Phi}^\top(\overline{s})\, \mathbf{\Phi}(\overline{s})\mathrm{d}\overline{s} = \left[\begin{array}{ccc|c} \mathbf{Z}_{xx} & \mathbf{0}_{n_x\times n_y} & \mathbf{0}_{n_x\times n_z} & \mathbf{0}_{n_x\times n_\alpha} \\ * & \mathbf{Z}_{yy} & \mathbf{0} & \mathbf{0}_{n_y\times n_\alpha} \\ * & * & \mathbf{Z}_{zz} & \mathbf{0}_{n_z\times n_\alpha} \\ \hline * & * & * & \mathbf{0}_{n_\alpha\times n_\alpha} \end{array}\right]$$

Secondly, the rotary inertias are based on the intermediate variable $\mathbf{Y}_{\psi\psi} \in \mathbb{R}^{n_f \times n_f}$:

$$\mathbf{Y}_{\psi\psi} \triangleq \int_0^l \rho\, \boldsymbol{\phi}_\psi(\overline{s})\boldsymbol{\phi}_\psi^\top(\overline{s})\, \mathrm{d}\overline{s} = \int_0^l \rho \left[\begin{array}{c|ccc} \mathbf{0}_{n_x\times n_x} & \mathbf{0}_{n_x\times n_y} & \mathbf{0}_{n_x\times n_z} & \mathbf{0}_{n_x\times n_\alpha} \\ \hline * & \boldsymbol{\phi}_y^{'}\boldsymbol{\phi}_y^{'\top} & \boldsymbol{\phi}_y^{'}\boldsymbol{\phi}_z^{'\top} & \boldsymbol{\phi}_y^{'}\boldsymbol{\phi}_\alpha^{\top} \\ * & * & \boldsymbol{\phi}_z^{'}\boldsymbol{\phi}_z^{'\top} & \boldsymbol{\phi}_z^{'}\boldsymbol{\phi}_\alpha^{\top} \\ * & * & * & \boldsymbol{\phi}_\alpha\boldsymbol{\phi}_\alpha^{\top} \end{array}\right] \mathrm{d}\overline{s}$$
$$= \left[\begin{array}{c|ccc} \mathbf{0}_{n_x\times n_x} & \mathbf{0}_{n_x\times n_y} & \mathbf{0}_{n_x\times n_z} & \mathbf{0}_{n_x\times n_\alpha} \\ \hline * & \mathbf{Y}_{yy} & \mathbf{Y}_{yz} & \mathbf{Y}_{y\alpha} \\ * & * & \mathbf{Y}_{zz} & \mathbf{Y}_{z\alpha} \\ * & * & * & \mathbf{Y}_{\alpha\alpha} \end{array}\right] \tag{B.23}$$

while its diagonal form considers only the cross-products in each different direction by:

$$\bar{\mathbf{Y}}_{\boldsymbol{\psi}\boldsymbol{\psi}} \triangleq \int_0^l \rho\, \boldsymbol{\Delta}^\top(\overline{s})\boldsymbol{\Delta}(\overline{s})\, \mathrm{d}\overline{s} = \left[\begin{array}{c|ccc} \mathbf{0}_{n_x\times n_x} & \mathbf{0}_{n_x\times n_y} & \mathbf{0}_{n_x\times n_z} & \mathbf{0}_{n_x\times n_\alpha} \\ \hline * & \mathbf{Y}_{yy} & \mathbf{0}_{n_y\times n_z} & \mathbf{0}_{n_y\times n_\alpha} \\ * & * & \mathbf{Y}_{zz} & \mathbf{0}_{n_z\times n_\alpha} \\ * & * & * & \mathbf{Y}_{\alpha\alpha} \end{array}\right] \tag{B.24}$$

Based on these notations, the rotary inertias are stored in $\bar{\mathbf{Z}}_{\boldsymbol{\psi}\boldsymbol{\psi}} \in \mathbb{R}^{n_f \times n_f}$, defined by:

$$\bar{\mathbf{Z}}_{\boldsymbol{\psi}\boldsymbol{\psi}} \triangleq \int_0^l \rho\, \boldsymbol{\Delta}^\top(\overline{s})\mathbf{J}_S\boldsymbol{\Delta}(\overline{s})\, \mathrm{d}\overline{s} = \left[\begin{array}{c|ccc} \mathbf{0}_{n_x\times n_x} & \mathbf{0}_{n_x\times n_y} & \mathbf{0}_{n_x\times n_z} & \mathbf{0}_{n_x\times n_\alpha} \\ \hline * & I_z\mathbf{Y}_{yy} & I_{yz}\mathbf{Y}_{yz} & \mathbf{0}_{n_y\times n_\alpha} \\ * & * & I_y\ \mathbf{Y}_{zz} & \mathbf{0}_{n_z\times n_\alpha} \\ * & * & * & I_p\mathbf{Y}_{\alpha\alpha} \end{array}\right] \tag{B.25}$$

where the terms of the previous matrix $\mathbf{Y}$ are weighted by the second moments of area, stored in:

$$\mathbf{J}_S = \iint_S -\left(\delta\mathbf{p}(y,z)^\times\right)^2 \mathrm{d}y\mathrm{d}z = \left[\begin{array}{ccc} I_p & 0 & 0 \\ * & I_y & I_{yz} \\ * & * & I_z \end{array}\right] \tag{B.26}$$

$$\mathbf{I}_S = \iint_S \delta\mathbf{p}(y,z)\,\delta\mathbf{p}(y,z)^\top \mathrm{d}y\mathrm{d}z = \left[\begin{array}{ccc} 0 & 0 & 0 \\ * & I_z & -I_{yz} \\ * & * & I_y \end{array}\right] \tag{B.27}$$

with $I_y = \iint_S z^2\, \mathrm{d}y\mathrm{d}z$ and $I_z = \iint_S y^2\, \mathrm{d}y\mathrm{d}z$ the second moments of area w.r.t. the transversal axis Y and Z, and $I_p = \iint_S (y^2+z^2)\mathrm{d}y\mathrm{d}z$ the polar second moment of area, verifying : $I_p = I_y + I_z$.

B.2.2.2 Flexible/Flexible Terms

The kinetic energy decomposition is now computed explicitly. The flexible/flexible terms T_{uu} and $T_{\psi\psi}$ are derived first, and then the rigid/flexible terms follow.

Contribution of T_{uu}

Developing T_{uu} using $\mathbf{v}_u$ in (B.16b), the following matrix form is obtained:

$$T_{\text{uu}} = \frac{1}{2}\int_{\mathcal{B}} \mathbf{v}_u^{\top}\,\mathbf{v}_u\,\mathrm{d}m = \frac{1}{2}\int_{\mathcal{B}} \mathbf{t}_{\text{o}}^{\top} \left[\begin{array}{cc|c} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n_f} \\ * & -\left(\boldsymbol{\Phi}\boldsymbol{\delta}\right)^{\times}\left(\boldsymbol{\Phi}\boldsymbol{\delta}\right)^{\times} & \left(\boldsymbol{\Phi}\,\boldsymbol{\delta}\right)^{\times}\boldsymbol{\Phi} \\ \hline * & * & \boldsymbol{\Phi}^{\top}\boldsymbol{\Phi} \end{array}\right] \mathbf{t}_{\text{o}}\,\mathrm{d}m$$

summarized by:

$$T_{\text{uu}} = \frac{1}{2}\,\mathbf{t}_{\text{o}}^{\top} \left[\begin{array}{cc|c} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n_f} \\ * & \mathbf{I}_{uu}(\boldsymbol{\delta},\boldsymbol{\delta}) & \mathbf{H}_{uu}(\boldsymbol{\delta}) \\ \hline * & * & \bar{\mathbf{Z}}_{\boldsymbol{uu}} \end{array}\right] \mathbf{t}_{\text{o}} \tag{B.28}$$

Matrix $\mathbf{I}_{uu} \in \mathbb{R}^{3\times3}$ is introduced to denote the flexible contribution of the translational modes on the second moment of inertia. It is defined by:

$$\begin{aligned} \mathbf{I}_{uu}(\boldsymbol{\delta}_1,\boldsymbol{\delta}_2) &= -\int_0^l \rho S\,\left(\boldsymbol{\Phi}\boldsymbol{\delta}_1\right)^{\times}\left(\boldsymbol{\Phi}\boldsymbol{\delta}_2\right)^{\times}\,\mathrm{d}\bar{s} \\ &= \Big(\boldsymbol{\delta}_1^{\top}\bar{\mathbf{Z}}_{\boldsymbol{uu}}\boldsymbol{\delta}_2\Big)\mathbf{I}_3 - \left[\boldsymbol{\delta}_1\right]_{\mathbf{u}}^{\top}\mathbf{Z}_{uu}\left[\boldsymbol{\delta}_2\right]_{\mathbf{u}} \end{aligned} \tag{B.29}$$

where the flexible variables in translation are arranged in a diagonal way through the notation:

$$\left[\boldsymbol{\delta}\right]_{\mathbf{u}} = \left[\begin{array}{ccc} \boldsymbol{\delta}_x & \mathbf{0}_{n_x\times1} & \mathbf{0}_{n_x\times1} \\ \mathbf{0}_{n_y\times1} & \boldsymbol{\delta}_y & \mathbf{0}_{n_y\times1} \\ \mathbf{0}_{n_z\times1} & \mathbf{0}_{n_z\times1} & \boldsymbol{\delta}_z \\ \hline \mathbf{0}_{n_\alpha\times1} & \mathbf{0}_{n_\alpha\times1} & \mathbf{0}_{n_\alpha\times1} \end{array}\right] \tag{B.30}$$

In addition, matrix $\mathbf{H}_{uu} \in \mathbb{R}^{3\times n_f}$ denotes the flexible contribution of $\boldsymbol{\phi}_u$ on the momentum. This influence is due to to the flexible angular rate produced when the section beam is rotating

due to bending. Derived from a cross-product adapted for block matrices, its expression reads:

$$\mathbf{H}_{uu}(\boldsymbol{\delta}) = \int_0^l \rho S \, (\boldsymbol{\Phi}\,\boldsymbol{\delta})^{\times}\,\boldsymbol{\Phi}\,\mathrm{d}\overline{s} = \left[\begin{array}{ccc|c} \mathbf{0}_{1\times n_x} & -\boldsymbol{\delta}_z^{\top}\mathbf{Z}_{zy} & \boldsymbol{\delta}_y^{\top}\mathbf{Z}_{yz} & \mathbf{0}_{1\times n_\alpha} \\ \boldsymbol{\delta}_z^{\top}\mathbf{Z}_{zx} & \mathbf{0}_{1\times n_y} & -\boldsymbol{\delta}_x^{\top}\mathbf{Z}_{xz} & \mathbf{0}_{1\times n_\alpha} \\ -\boldsymbol{\delta}_y^{\top}\mathbf{Z}_{yx} & \boldsymbol{\delta}_x^{\top}\mathbf{Z}_{xy} & \mathbf{0}_{1\times n_z} & \mathbf{0}_{1\times n_\alpha} \end{array}\right] \tag{B.31}$$

Contribution of $T_{\psi\psi}$

Due to the hypothesis of a slender segment, the cross-sectional dimensions are small compared to its length. Therefore, the *rotary inertias* denoted by $\mathbf{I}_{\psi\psi}$, $\mathbf{H}_{\psi\psi}$ and $\bar{\mathbf{Z}}_{\boldsymbol{\psi\psi}}$ are introduced for sake of completeness, but remains negligible compared to the *translational inertias* introduced by $\mathbf{I}_{uu}$, $\mathbf{H}_{uu}$ and $\bar{\mathbf{Z}}_{\boldsymbol{uu}}$. They could be omitted assuming that the second moments of area I_y, I_z, I_{yz} and I_p are much smaller than the cross-section area S (see (Timoshenko, 1955; Cyril, 1988)).

Developing $T_{\psi\psi}$ using (B.16c), the same kind of matrix form as (B.28) appears:

$$T_{\psi\psi} = \frac{1}{2}\int_{\mathcal{B}} \mathbf{v}_{\psi}^{\top}\,\mathbf{v}_{\psi}\,\mathrm{d}m = \frac{1}{2}\int_{\mathcal{B}} \mathbf{t}_{\mathrm{o}}^{\top} \left[\begin{array}{cc|c} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times n_f} \\ * & -\left(\delta\mathbf{p}^{\times}\left(\boldsymbol{\Delta}\,\boldsymbol{\delta}\right)\right)^{\times,2} & \left(\delta\mathbf{p}^{\times}\left(\boldsymbol{\Delta}\,\boldsymbol{\delta}\right)\right)^{\times}\delta\mathbf{p}^{\times}\boldsymbol{\Delta} \\ \hline * & * & -\boldsymbol{\Delta}^{\top}\delta\mathbf{p}^{\times}\delta\mathbf{p}^{\times}\boldsymbol{\Delta} \end{array}\right] \mathbf{t}_{\mathrm{o}}\,\mathrm{d}m$$

summarized by:

$$T_{\psi\psi} = \frac{1}{2}\,\mathbf{t}_{\mathrm{o}}^{\top} \left[\begin{array}{cc|c} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times n_f} \\ * & \mathbf{I}_{\psi\psi}(\boldsymbol{\delta},\boldsymbol{\delta}) & \mathbf{H}_{\psi\psi}(\boldsymbol{\delta}) \\ \hline * & * & \bar{\mathbf{Z}}_{\boldsymbol{\psi\psi}} \end{array}\right] \mathbf{t}_{\mathrm{o}} \tag{B.32}$$

A rotary corrective term is added on the second moment of inertia with matrix $\mathbf{I}_{\psi\psi} \in \mathbb{R}^{3\times 3}$:

$$\begin{aligned} \mathbf{I}_{\psi\psi}(\boldsymbol{\delta}_1,\boldsymbol{\delta}_2) &= -\int_{\mathcal{B}} \rho \left(\delta\mathbf{p}^{\times}\left(\boldsymbol{\Delta}\,\boldsymbol{\delta}_1\right)\right)^{\times}\left(\delta\mathbf{p}^{\times}\left(\boldsymbol{\Delta}\,\boldsymbol{\delta}_2\right)\right)^{\times}\,\mathrm{d}y\mathrm{d}z\mathrm{d}\overline{s} \\ &= \left(\left(\boldsymbol{\delta}_1^{\top}\bar{\mathbf{Y}}_{\boldsymbol{\psi\psi}}\boldsymbol{\delta}_2\right)\mathbf{I}_3 - \left[\boldsymbol{\delta}_1\right]_{\boldsymbol{\psi}}^{\top}\mathbf{Y}_{\psi\psi}\left[\boldsymbol{\delta}_2\right]_{\boldsymbol{\psi}}\right)\mathbf{I}_S + \mathbf{J}_S\left(\left[\boldsymbol{\delta}_1\right]_{\boldsymbol{\psi}}^{\top}\mathbf{Y}[\psi\psi]\left[\boldsymbol{\delta}_2\right]_{\boldsymbol{\psi}}\right) \end{aligned} \tag{B.33}$$

where the flexible variables in rotation are arranged in an anti-diagonal way through the

notation:

$$[\boldsymbol{\delta}]_{\psi} = \left[\begin{array}{ccc} \mathbf{0}_{n_x\times 1} & \mathbf{0}_{n_x\times 1} & \mathbf{0}_{n_x\times 1} \\ \hline \mathbf{0}_{n_y\times 1} & \mathbf{0}_{n_y\times 1} & \boldsymbol{\delta}_y \\ \mathbf{0}_{n_z\times 1} & \boldsymbol{\delta}_z & \mathbf{0}_{n_z\times 1} \\ \boldsymbol{\delta}_\alpha & \mathbf{0}_{n_\alpha\times 1} & \mathbf{0}_{n_\alpha\times 1} \end{array}\right] \tag{B.34}$$

Finally, the rotary contribution on the angular momentum $\mathbf{H}_{\psi\psi} \in \mathbb{R}^{3\times n_f}$ is given by:

$$\mathbf{H}_{\psi\psi}(\boldsymbol{\delta}) = \mathbf{I}_S \left[\begin{array}{c|ccc} \mathbf{0}_{1\times n_x} & \boldsymbol{\delta}_z^\top \mathbf{Y}_{zy} & -\boldsymbol{\delta}_y^\top \mathbf{Y}_{yz} & \mathbf{0}_{1\times n_\alpha} \\ \mathbf{0}_{1\times n_x} & -\boldsymbol{\delta}_\alpha^\top \mathbf{Y}_{\alpha y} & \mathbf{0}_{1\times n_z} & \boldsymbol{\delta}_y^\top \mathbf{Y}_{y\alpha} \\ \mathbf{0}_{1\times n_x} & \mathbf{0}_{1\times n_y} & \boldsymbol{\delta}_\alpha^\top \mathbf{Y}_{\alpha z} & -\boldsymbol{\delta}_z^\top \mathbf{Y}_{z\alpha} \end{array}\right] \tag{B.35}$$

B.2.2.3 Rigid/Flexible Cross Terms

The kinetic cross terms $T_{\mathrm{ru}} + T_{\mathrm{ur}}$, $T_{\mathrm{r}\psi} + T_{\psi\mathrm{r}}$ and $T_{\mathrm{u}\psi} + T_{\psi\mathrm{u}}$ are computed using their symmetry property: $T_{\mathrm{ij}} = T_{\mathrm{ji}}$. In the sequel, it is worth noticing that the centroidal axis is, by definition, passing through the "center of area" $C_{\mathcal{A}}$ of each cross-section, that is denoted C in the thesis. It implies that $C = C_{\mathcal{A}}$, by definition of C. Therefore the following integral reduces to zero:

$$\iint_S \delta\mathbf{p}(y,z)\,\mathrm{d}y\mathrm{d}z = S\,\delta\mathbf{p}_{C_{\mathcal{A}}} = \mathbf{0}_{3\times 1} \tag{B.36}$$

Thus, the decomposition of the relative position of any point P by $\mathbf{p} = \mathbf{p}_c + \delta\mathbf{p}$ in (5.5), allows to simplify the following integral arising in the computation of the kinetic cross terms:

$$\begin{aligned} \int_{\mathcal{B}} \mathbf{p}\,\mathrm{d}m &= \int \rho \left(\iint_S (\mathbf{p}_c(s) + \delta\mathbf{p}(y,z))\mathrm{d}y\mathrm{d}z \right) \mathrm{d}s \\ &= \int \rho\, S\, \mathbf{p}_c(s)\,\mathrm{d}s \end{aligned}$$

Computation of T_{ru}

The matrix form of $T_{\mathrm{ru}} + T_{\mathrm{ur}}$ is inferred using (B.16a) and (B.16b).

$$\begin{aligned} T_{\mathrm{ru}} + T_{\mathrm{ur}} &= \frac{1}{2} \int_{\mathcal{B}} \left(\mathbf{v}_r^\top \mathbf{v}_u + \mathbf{v}_u^\top \mathbf{v}_r \right) \mathrm{d}m \\ &= \frac{1}{2} \int_{\mathcal{B}} \mathbf{t}_{\mathrm{o}}^\top \left[\begin{array}{ccc|c} \mathbf{0}_{3\times 3} & -\left(\boldsymbol{\Phi}\,\boldsymbol{\delta}\right)^\times & & \boldsymbol{\Phi} \\ * & -\mathbf{p}^\times \left(\boldsymbol{\Phi}\,\boldsymbol{\delta}\right)^\times - \left(\boldsymbol{\Phi}\,\boldsymbol{\delta}\right)^\times \mathbf{p}^\times & & \mathbf{p}^\times \boldsymbol{\Phi} \\ \hline * & * & & \mathbf{0}_{n_f\times n_f} \end{array}\right] \mathbf{t}_{\mathrm{o}}\,\mathrm{d}m \end{aligned}$$

summarized by:

$$T_{\mathrm{ru}} + T_{\mathrm{ur}} = \frac{1}{2}\mathbf{t}_{\mathrm{o}}^{\top} \left[\begin{array}{cc|c} \mathbf{0}_{3\times3} & -\left(\mathbf{V}_u\boldsymbol{\delta}\right)^{\times} & \mathbf{P}_{ru} \\ * & \mathbf{I}_{ru}(\boldsymbol{\delta}) & \mathbf{H}_{ru} \\ \hline * & * & \mathbf{0}_{n_f\times n_f} \end{array}\right] \mathbf{t}_{\mathrm{o}} \tag{B.37}$$

The same notations are used to denote the contributions on the flexible linear and angular momentum with $\mathbf{P}$ and $\mathbf{H}$, as well as the corrective term on the second moment of inertia with $\mathbf{I}$. They are defined by:

$$\mathbf{P}_{ru} = \mathbf{V}_u \tag{B.38}$$

$$\mathbf{I}_{ru}(\boldsymbol{\delta}) = -\langle \mathbf{p}_{\mathrm{a}}, \mathbf{V}_u\boldsymbol{\delta}\rangle - \langle \mathbf{e}_x, \mathbf{W}_u\boldsymbol{\delta}\rangle \tag{B.39}$$

$$\mathbf{H}_{ru} = \mathbf{p}_{\mathrm{a}}^{\times}\mathbf{V}_u + \mathbf{e}_x^{\times}\mathbf{W}_u \tag{B.40}$$

where the product $\langle .,. \rangle$ is defined by: $\langle u, v\rangle = u^{\times}\, v^{\times} + v^{\times}\, u^{\times}$. Besides, the vector $\mathbf{p}_{\mathrm{a}}$ denotes the rigid part of the segment, from its base to the beginning of the slender flexible part, as illustrated in Fig. B.2. On the other hand, the vector $\mathbf{e}_x$ denotes the axis of the slender part (i.e., the x axis in the classical beam theory, but expressed here in $\mathcal{R}_{\mathrm{o}}$).

It can be noticed that $\mathbf{V}_u \in \mathbb{R}^{3\times n_f}$ provides the flexible shift of the CoM location when the segment is in a deformed state: $\mathbf{c}_f = \mathbf{c} + \mathbf{V}_u\boldsymbol{\delta}$. Similarly, $\mathbf{W}_u \in \mathbb{R}^{3\times n_f}$ is a corrective term on the angular momentum due to the translational modes.

Computation of $T_{\mathrm{r}\psi}$

In the same way, $T_{\mathrm{r}\psi} + T_{\psi\mathrm{r}}$ is derived using (B.16a) and (B.16c), to obtain:

$$\begin{aligned} T_{\mathrm{r}\psi} + T_{\psi\mathrm{r}} &= \frac{1}{2}\int_{\mathcal{B}} \left(\mathbf{v}_r^{\top}\mathbf{v}_{\psi} + \mathbf{v}_{\psi}^{\top}\mathbf{v}_r\right) \mathrm{d}m \\ &= \frac{1}{2}\int_{\mathcal{B}} \mathbf{t}_{\mathrm{o}}^{\top} \left[\begin{array}{ccc|c} \mathbf{0}_{3\times3} & \left(\delta\mathbf{p}^{\times}(\boldsymbol{\Delta\delta})\right)^{\times} & & -\delta\mathbf{p}^{\times}\boldsymbol{\Delta} \\ * & \mathbf{p}^{\times}\left(\delta\mathbf{p}^{\times}(\boldsymbol{\Delta\delta})\right)^{\times} + \left(\delta\mathbf{p}^{\times}(\boldsymbol{\Delta\delta})\right)^{\times}\mathbf{p}^{\times} & & -\mathbf{p}^{\times}\,\delta\mathbf{p}^{\times}\boldsymbol{\Delta} \\ \hline * & * & & \mathbf{0}_{n_f\times n_f} \end{array}\right] \mathbf{t}_{\mathrm{o}}\, \mathrm{d}m \end{aligned}$$

summarized by:

$$T_{\mathrm{r}\psi} + T_{\psi\mathrm{r}} = \frac{1}{2}\mathbf{t}_{\mathrm{o}}^{\top} \left[\begin{array}{cc|c} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n_f} \\ * & \mathbf{I}_{r\psi}(\boldsymbol{\delta}) & \mathbf{H}_{r\psi} \\ \hline * & * & \mathbf{0}_{n_f\times n_f} \end{array}\right] \mathbf{t}_{\mathrm{o}} \tag{B.41}$$

The hypothesis on the center of area in (B.36) has been used to cancel the upper terms, while the remaining ones still influence the angular momentum with matrices:

$$\mathbf{I}_{r\psi}(\boldsymbol{\delta}) = (\mathbf{V}_{\psi}\boldsymbol{\delta})^{\times}\ \mathbf{J}_S - \mathbf{J}_S\ (\mathbf{V}_{\psi}\boldsymbol{\delta})^{\times} \tag{B.42}$$

$$\mathbf{H}_{r\psi} = \mathbf{J}_S\mathbf{V}_{\psi} \tag{B.43}$$

Computation of $T_{\mathrm{u}\psi}$

The matrix form of $T_{\mathrm{u}\psi} + T_{\psi\mathrm{u}}$ is inferred using (B.16b) and (B.16c).

$$\begin{aligned} T_{\mathrm{u}\psi} + T_{\psi\mathrm{u}} &= \frac{1}{2}\int_{\mathcal{B}} \left(\mathbf{v}_u^{\top}\mathbf{v}_{\psi} + \mathbf{v}_{\psi}^{\top}\mathbf{v}_u\right) \mathrm{d}m \\ &= \frac{1}{2}\int_{\mathcal{B}} \mathbf{t}_{\mathrm{o}}^{\top} \left[\begin{array}{cc|c} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n_f} \\ * & (\boldsymbol{\Phi}\,\boldsymbol{\delta})^{\times}\ (\delta\mathbf{p}^{\times}(\boldsymbol{\Delta}\boldsymbol{\delta}))^{\times} & -(\boldsymbol{\Phi}\,\boldsymbol{\delta})^{\times}\ \delta\mathbf{p}^{\times}\boldsymbol{\Delta} \\ & \ldots + (\delta\mathbf{p}^{\times}(\boldsymbol{\Delta}\boldsymbol{\delta}))^{\times}\ (\boldsymbol{\Phi}\,\boldsymbol{\delta})^{\times} & \\ \hline * & * & -\boldsymbol{\Phi}^{\top}\,\delta\mathbf{p}^{\times}\boldsymbol{\Delta} \\ & & \ldots + \boldsymbol{\Delta}^{\top}\delta\mathbf{p}^{\times}\,\boldsymbol{\Phi} \end{array}\right] \mathbf{t}_{\mathrm{o}}\,\mathrm{d}m \end{aligned}$$

Noticing that $\mathbf{v}_u$ does not contain any term depending upon $\delta\mathbf{p}(y,z)$, the integration of $\mathbf{v}_{\psi}$ in $T_{\mathrm{u}\psi} + T_{\psi\mathrm{u}}$ over the cross-section domain vanishes thanks to the center-of-area hypothesis in (B.36). It leads to the following result:

$$T_{\mathrm{u}\psi} + T_{\psi\mathrm{u}} = 0 \tag{B.44}$$

B.2.3 Global Kinetic Energy

Finally, equations (B.18), (B.28), (B.32), (B.37), (B.41) and (B.44) are combined to derive the full expression of the kinetic energy given in (5.16).

$$T = \frac{1}{2}\,\mathbf{t}_{\mathrm{o}}^{\top}\,\mathbf{M}_{\mathrm{f}}\,\mathbf{t}_{\mathrm{o}} \tag{B.45}$$

By gathering all the previous results, the flexible mass matrix is given by:

$$\mathbf{M}_{\mathrm{f}} = \left[\begin{array}{ccc|c} \mathbf{M}_{vv} & \mathbf{M}_{v\omega} & & \mathbf{M}_{v\delta} \\ * & \mathbf{M}_{\omega\omega} & & \mathbf{M}_{\omega\delta} \\ \hline * & * & & \mathbf{M}_{\delta\delta} \end{array}\right] \tag{B.46}$$

with

$$\begin{aligned} \mathbf{M}_{vv} &= m\mathbf{I}_3 \\ \mathbf{M}_{v\omega} &= -\left(\mathbf{c} + \mathbf{V}_u\boldsymbol{\delta}\right)^{\times} \\ \mathbf{M}_{v\delta} &= \mathbf{P}_{ru} \\ \mathbf{M}_{\omega\omega} &= \mathbf{I}_{\mathrm{o}} + \mathbf{I}_{ru}(\boldsymbol{\delta}) + \mathbf{I}_{r\psi}(\boldsymbol{\delta}) + \mathbf{I}_{uu}(\boldsymbol{\delta},\boldsymbol{\delta}) + \mathbf{I}_{\psi\psi}(\boldsymbol{\delta},\boldsymbol{\delta}) \\ \mathbf{M}_{\omega\delta} &= \mathbf{H}_{ru} + \mathbf{H}_{r\psi} + \mathbf{H}_{uu}(\boldsymbol{\delta}) + \mathbf{H}_{\psi\psi}(\boldsymbol{\delta}) \\ \mathbf{M}_{\delta\delta} &= \bar{\mathbf{Z}}_{\boldsymbol{uu}} + \bar{\mathbf{Z}}_{\boldsymbol{\psi\psi}} \end{aligned}$$

where matrices $\mathbf{P}_{ii}$, $\mathbf{H}_{ii}$, and $\mathbf{I}_{ii}$, contribute to, respectively, the flexible linear and angular momentums, and to the inertia tensor. Their exact expressions are given as follows, to summarize the previous computations:

$$\begin{aligned} \mathbf{P}_{ru} &= \mathbf{V}_u \\ \mathbf{I}_{ru}(\boldsymbol{\delta}) &= -\langle \mathbf{p}_{\mathrm{a}}, \mathbf{V}_u\boldsymbol{\delta}\rangle - \langle \mathbf{e}_x, \mathbf{W}_u\boldsymbol{\delta}\rangle \\ \mathbf{I}_{r\psi}(\boldsymbol{\delta}) &= \left(\mathbf{V}_\psi\boldsymbol{\delta}\right)^{\times}\mathbf{J}_S - \mathbf{J}_S\left(\mathbf{V}_\psi\boldsymbol{\delta}\right)^{\times} \\ \mathbf{I}_{uu}(\boldsymbol{\delta}_1,\boldsymbol{\delta}_2) &= \left(\boldsymbol{\delta}_1^{\top}\bar{\mathbf{Z}}_{\boldsymbol{uu}}\boldsymbol{\delta}_2\right)\mathbf{I}_3 - [\boldsymbol{\delta}_1]_{\mathbf{u}}^{\top}\mathbf{Z}_{uu}[\boldsymbol{\delta}_2]_{\mathbf{u}} \\ \mathbf{I}_{\psi\psi}(\boldsymbol{\delta}_1,\boldsymbol{\delta}_2) &= \left(\left(\boldsymbol{\delta}_1^{\top}\bar{\mathbf{Y}}_{\boldsymbol{\psi\psi}}\boldsymbol{\delta}_2\right)\mathbf{I}_3 - [\boldsymbol{\delta}_1]_{\boldsymbol{\psi}}^{\top}\mathbf{Y}_{\psi\psi}[\boldsymbol{\delta}_2]_{\boldsymbol{\psi}}\right)\mathbf{I}_S + \mathbf{J}_S\left([\boldsymbol{\delta}_1]_{\boldsymbol{\psi}}^{\top}\mathbf{Y}[\psi\psi][\boldsymbol{\delta}_2]_{\boldsymbol{\psi}}\right) \\ \mathbf{H}_{ru} &= \mathbf{p}_{\mathrm{a}}^{\times}\mathbf{V}_u + \mathbf{e}_x^{\times}\mathbf{W}_u \\ \mathbf{H}_{r\psi} &= \mathbf{J}_S\mathbf{V}_\psi \\ \mathbf{H}_{uu}(\boldsymbol{\delta}) &= \left[\begin{array}{ccc|c} \mathbf{0}_{1\times n_x} & -\boldsymbol{\delta}_z^{\top}\mathbf{Z}_{zy} & \boldsymbol{\delta}_y^{\top}\mathbf{Z}_{yz} & \mathbf{0}_{1\times n_\alpha} \\ \boldsymbol{\delta}_z^{\top}\mathbf{Z}_{zx} & \mathbf{0}_{1\times n_y} & -\boldsymbol{\delta}_x^{\top}\mathbf{Z}_{xz} & \mathbf{0}_{1\times n_\alpha} \\ -\boldsymbol{\delta}_y^{\top}\mathbf{Z}_{yx} & \boldsymbol{\delta}_x^{\top}\mathbf{Z}_{xy} & \mathbf{0}_{1\times n_z} & \mathbf{0}_{1\times n_\alpha} \end{array}\right] \\ \mathbf{H}_{\psi\psi}(\boldsymbol{\delta}) &= \mathbf{I}_S\left[\begin{array}{c|ccc} \mathbf{0}_{1\times n_x} & \boldsymbol{\delta}_z^{\top}\mathbf{Y}_{zy} & -\boldsymbol{\delta}_y^{\top}\mathbf{Y}_{yz} & \mathbf{0}_{1\times n_\alpha} \\ \mathbf{0}_{1\times n_x} & -\boldsymbol{\delta}_\alpha^{\top}\mathbf{Y}_{\alpha y} & \mathbf{0}_{1\times n_z} & \boldsymbol{\delta}_y^{\top}\mathbf{Y}_{y\alpha} \\ \mathbf{0}_{1\times n_x} & \mathbf{0}_{1\times n_y} & \boldsymbol{\delta}_\alpha^{\top}\mathbf{Y}_{\alpha z} & -\boldsymbol{\delta}_z^{\top}\mathbf{Y}_{z\alpha} \end{array}\right] \end{aligned}$$

B.3 Quasi-Lagrangian Equations

This section aims at developing the nonlinear terms arising in the derivation of flexible dynamics by the quasi-Lagrangian equations in (5.21). By computing them explicitly, it is possible to identify the terms that must be updated on-line during the simulation from those that can be computed and stored off-line. The time derivative of the mass matrix is obtained first, and then the state derivatives of the Lagrangian are obtained.

Reminder: The system Lagrangian is given under the following matrix form using the kinetic energy in (5.16), the potential energy in (5.20), and replacing them in the general Lagrangian in (4.10). Denoting $\mathbf{M}_\mathrm{f}$ the mass matrix and $\mathbf{K}_\mathrm{f}$ the stiffness matrix, it reads:

$$L = \underbrace{\frac{1}{2}\,\mathbf{t}_\mathrm{o}^\top\,\mathbf{M}(\boldsymbol{\delta})\,\mathbf{t}_\mathrm{o}}_{T} - \underbrace{\frac{1}{2}\,\mathbf{x}_\mathrm{o}^\top\,\mathbf{K}_\mathrm{f}\,\mathbf{x}_\mathrm{o}}_{U} \tag{B.47}$$

The flexible dynamics were obtained by the hybrid state equations, written as follows in (5.22):

$$\boldsymbol{Q} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \mathbf{t}_\mathrm{o}}\right) - \frac{\partial \tilde{L}}{\partial \mathbf{x}_\mathrm{o}} \tag{B.48}$$

with the extended state derivative vector given by (5.23):

$$\frac{\partial \tilde{L}}{\partial \mathbf{x}_\mathrm{o}} = \begin{bmatrix} \mathbf{0}_{3\times 1} \\ -\boldsymbol{v}^\times\left(\frac{\partial L}{\partial \boldsymbol{v}_\mathrm{o}}\right) \\ \frac{\partial L}{\partial \boldsymbol{\delta}} \end{bmatrix} = -\underbrace{\begin{bmatrix} \mathbf{0}_{3\times 1} \\ \boldsymbol{v}_\mathrm{o}^\times\left(\frac{\partial L}{\partial \boldsymbol{v}_\mathrm{o}}\right) \\ -\frac{\partial}{\partial \boldsymbol{\delta}}\left(T\right) \end{bmatrix}}_{=\tilde{\mathbf{M}}_\mathrm{f}(\boldsymbol{\delta},\boldsymbol{\omega}_\mathrm{o})\,\mathbf{t}_\mathrm{o}} - \underbrace{\begin{bmatrix} \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 1} \\ \frac{\partial}{\partial \boldsymbol{\delta}}\left(U\right) \end{bmatrix}}_{=\mathbf{K}_\mathrm{f}\mathbf{x}_\mathrm{o}} \tag{B.49}$$

This section aims at providing the explicit expressions of matrices $\mathring{\mathbf{M}}_\mathrm{f}$ and $\tilde{\mathbf{M}}_\mathrm{f}$ to derive the flexible dynamics as (5.26):

$$\boldsymbol{Q} = \mathbf{M}_\mathrm{f}\,\dot{\mathbf{t}}_\mathrm{o} + \left(\mathring{\mathbf{M}}_\mathrm{f} + \boldsymbol{\Omega}_\mathrm{o}\,\mathbf{M}_\mathrm{f} + \tilde{\mathbf{M}}_\mathrm{f}\right)\mathbf{E}_v\,\mathbf{t}_\mathrm{o} + \mathbf{K}_\mathrm{f}\,\mathbf{x}_\mathrm{o} \tag{B.50}$$

B.3.1 Lagrangian time derivative

The time derivative of $\left(\frac{\partial L}{\partial \mathbf{t}_\mathrm{o}}\right)$ in (B.48) is performed with respect to the inertial frame. Based on Lagrangian expression in (B.47), this term reads:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \mathbf{t}_\mathrm{o}}\right) = \frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{M}_\mathrm{f}\,\mathbf{t}_\mathrm{o}\right) = \mathbf{M}_\mathrm{f}\,\dot{\mathbf{t}}_\mathrm{o} + \dot{\mathbf{M}}_\mathrm{f}\,\mathbf{t}_\mathrm{o}$$

Using the *vectrix* time derivative (Hughes, 1986), the term $\dot{\mathbf{M}}_\mathrm{f}$ is expanded to yield:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \mathbf{t}_\mathrm{o}}\right) = \mathbf{M}_\mathrm{f}\,\dot{\mathbf{t}}_\mathrm{o} + \left(\mathring{\mathbf{M}}_\mathrm{f} + \boldsymbol{\Omega}_\mathrm{o}\,\mathbf{M}_\mathrm{f} - \mathbf{M}_\mathrm{f}\,\boldsymbol{\Omega}_\mathrm{o}\right)\,\mathbf{t}_\mathrm{o} \tag{B.51}$$

where it is recalled that an inertial differentiation w.r.t. $\mathcal{R}_\mathrm{I}$ is denoted by $\dot{\mathbf{x}}$, while a local differentiation w.r.t. $\mathcal{R}_\mathrm{o}$ is denoted by $\mathring{\mathbf{x}}$. The augmented skew-symmetric matrix $\boldsymbol{\Omega}$ is introduced as:

$$\boldsymbol{\Omega}_\mathrm{o} = \left[\begin{array}{cc|c} \boldsymbol{\omega}_\mathrm{o}^\times & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times n_f} \\ * & \boldsymbol{\omega}_\mathrm{o}^\times & \mathbf{0}_{3\times n_f} \\ \hline * & * & \mathbf{0}_{n_f\times n_f} \end{array}\right] \tag{B.52}$$

The advantage of this formulation is that $\mathring{\mathbf{M}}_\mathrm{f}$ can be expressed more easily, since it only depends upon $\boldsymbol{\delta}(t)$ w.r.t. $\mathcal{R}_\mathrm{o}$ in (B.46). This time derivative is performed by evaluating the functions with a linear dependency $f(\boldsymbol{\delta})$ for $f(\dot{\boldsymbol{\delta}})$ (i.e., for terms $\mathbf{V}_u\boldsymbol{\delta}$, $\mathbf{I}_{ru}$, $\mathbf{I}_{r\psi}$, $\mathbf{H}_{uu}$ and $\mathbf{H}_{\psi\psi}$), and the functions with a quadratic dependency $f(\boldsymbol{\delta},\boldsymbol{\delta})$ for $f(\dot{\boldsymbol{\delta}},\boldsymbol{\delta}) + f(\boldsymbol{\delta},\dot{\boldsymbol{\delta}})$ (i.e., for terms $\mathbf{I}_{uu}$ and $\mathbf{I}_{\psi\psi}$). Using the mass matrix decomposition in (5.18), this time derivative is written as:

$$\mathring{\mathbf{M}}_\mathrm{f} = \mathbf{M}_\mathrm{f}^{(1)}(\dot{\boldsymbol{\delta}}) + \mathbf{M}_\mathrm{f}^{(2)}(\boldsymbol{\delta},\dot{\boldsymbol{\delta}}) \tag{B.53}$$

with

$$\mathbf{M}_\mathrm{f}^{(1)} = \left[\begin{array}{ccc} \mathbf{0}_{3\times 3} & \left(-\mathbf{V}_u\dot{\boldsymbol{\delta}}\right)^\times & \mathbf{0}_{3\times 3} \\ * & \mathbf{I}_{ru}(\dot{\boldsymbol{\delta}}) + \mathbf{I}_{r\psi}(\dot{\boldsymbol{\delta}}) & \mathbf{H}_{uu}(\dot{\boldsymbol{\delta}}) + \mathbf{H}_{\psi\psi}(\dot{\boldsymbol{\delta}}) \\ \hline * & * & \mathbf{0}_{n_f\times n_f} \end{array}\right]$$

$$\mathbf{M}_\mathrm{f}^{(2)} = \left[\begin{array}{ccc} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times n_f} \\ * & \mathbf{I}_{uu}(\dot{\boldsymbol{\delta}},\boldsymbol{\delta}) + \mathbf{I}_{uu}(\boldsymbol{\delta},\dot{\boldsymbol{\delta}}) + \mathbf{I}_{\psi\psi}(\dot{\boldsymbol{\delta}},\boldsymbol{\delta}) + \mathbf{I}_{\psi\psi}(\boldsymbol{\delta},\dot{\boldsymbol{\delta}}) & \mathbf{0}_{3\times n_f} \\ \hline * & * & \mathbf{0}_{n_f\times n_f} \end{array}\right]$$

B.3.2 Lagrangian state derivative

According to (B.46) in Appendix B.2, the only dependency of T upon the state variables $\mathbf{x}_\mathrm{o}$ is only through the $\boldsymbol{\delta}$-dependency of $\mathbf{M}_\mathrm{f}$. Regarding the potential energy U in (B.47), the state derivative is straightforward and yields $\mathbf{K}_\mathrm{f}\,\mathbf{x}_\mathrm{o}$. Therefore, the extended state derivative of the Lagrangian $\frac{\partial \tilde{L}}{\partial \mathbf{x}_\mathrm{o}}$ is divided into $\tilde{\mathbf{M}}_\mathrm{f}\,\mathbf{t}_\mathrm{o}$ and $\mathbf{K}_\mathrm{f}\,\mathbf{x}_\mathrm{o}$ in (B.49). The terms of $\tilde{\mathbf{M}}_\mathrm{f}$ are now developed according to the explicit expression of $\mathbf{M}_\mathrm{f}$ in (B.46).

Development of $\boldsymbol{v}_\mathrm{o}^\times \frac{\partial L}{\partial \boldsymbol{v}_\mathrm{o}}$: The second row of the state derivative is computed based on the first row of the mass matrix by:

$$\begin{aligned}
\boldsymbol{v}_\mathrm{o}^\times \frac{\partial L}{\partial \boldsymbol{v}_\mathrm{o}} &= \boldsymbol{v}_\mathrm{o}^\times \left(\begin{bmatrix} \mathbf{M}_{vv} & \mathbf{M}_{v\omega} & \mathbf{M}_{v\delta} \end{bmatrix} \mathbf{t}_\mathrm{o} \right) \\
&= \boldsymbol{v}_\mathrm{o}^\times \left(\begin{bmatrix} m\mathbf{I}_3 & -(\mathbf{c}+\mathbf{V}_u\boldsymbol{\delta})^\times & \mathbf{V}_u \end{bmatrix} \mathbf{t}_\mathrm{o} \right) \\
&= \boldsymbol{v}_\mathrm{o}^\times \left(m\boldsymbol{v}_\mathrm{o} - (\mathbf{c}+\mathbf{V}_u\boldsymbol{\delta})^\times\, \boldsymbol{\omega}_\mathrm{o} + \mathbf{V}_u\dot{\boldsymbol{\delta}} \right) \\
&= m\, \underbrace{\boldsymbol{v}_\mathrm{o} \times \boldsymbol{v}_\mathrm{o}}_{=0} + \left((\mathbf{c}+\mathbf{V}_u\boldsymbol{\delta})^\times \boldsymbol{\omega}_\mathrm{o} \right) \times \boldsymbol{v}_\mathrm{o} - \left(\mathbf{V}_u\dot{\boldsymbol{\delta}} \right) \times \boldsymbol{v}_\mathrm{o} \\
&= \left((\mathbf{c}+\mathbf{V}_u\boldsymbol{\delta})^\times \boldsymbol{\omega}_\mathrm{o}^\times - \boldsymbol{\omega}_\mathrm{o}^\times (\mathbf{c}+\mathbf{V}_u\boldsymbol{\delta})^\times \right) \boldsymbol{v}_\mathrm{o} - \left(\mathbf{V}_u\dot{\boldsymbol{\delta}} \right)^\times \boldsymbol{v}_\mathrm{o}
\end{aligned}$$

where one uses the Jacobi relation at the final step to develop the double cross-product. Coming back to the definition of the mass matrix terms, one eventually obtains:

$$\begin{aligned}
\boldsymbol{v}_\mathrm{o}^\times \frac{\partial L}{\partial \boldsymbol{v}_\mathrm{o}} &= \left(-\mathbf{M}_{v\omega}\boldsymbol{\omega}_\mathrm{o}^\times + \boldsymbol{\omega}_\mathrm{o}^\times \mathbf{M}_{v\omega} \right) \boldsymbol{v}_\mathrm{o} + \mathring{\mathbf{M}}_{v\omega}\boldsymbol{v}_\mathrm{o} \\
&= \left(\mathring{\mathbf{M}}_{v\omega} + \boldsymbol{\omega}_\mathrm{o}^\times \mathbf{M}_{v\omega} - \mathbf{M}_{v\omega}\boldsymbol{\omega}_\mathrm{o}^\times \right) \boldsymbol{v}_\mathrm{o}
\end{aligned}$$

which can also be re-written as:

$$\boldsymbol{v}_\mathrm{o}^\times \frac{\partial L}{\partial \boldsymbol{v}_\mathrm{o}} = \dot{\mathbf{M}}_{v\omega}\boldsymbol{v}_\mathrm{o} \tag{B.54}$$

This result will be helpful to simplify the global dynamics by canceling $\dot{\mathbf{M}}_{v\omega}$ in the inertial time derivative of the whole mass matrix.

Development of $\frac{\partial T}{\partial \boldsymbol{\delta}}$: The derivative of the kinetic energy with respect to the generalized coordinates $\mathbf{x}_\mathrm{o}$ reduces to the differentiation w.r.t. $\boldsymbol{\delta}$ only, since it does not depend explicitly

upon $\mathbf{r}_\mathrm{o}$, nor $\mathbf{\Psi}_\mathrm{o}$. The computation reads as follows:

$$\begin{aligned}\frac{\partial T}{\partial \boldsymbol{\delta}} &= \frac{1}{2}\,\frac{\partial}{\partial \boldsymbol{\delta}}\left(\mathbf{t}_\mathrm{o}^\top\,\mathbf{M}_\mathrm{f}(\boldsymbol{\delta})\,\mathbf{t}_\mathrm{o}\right)\\ &= \frac{1}{2}\,\frac{\partial}{\partial \boldsymbol{\delta}}\left(\mathbf{t}_\mathrm{o}^\top\,\left(\mathbf{M}_\mathrm{f}^{(1)}(\boldsymbol{\delta})+\mathbf{M}_\mathrm{f}^{(2)}(\boldsymbol{\delta},\boldsymbol{\delta})\right)\mathbf{t}_\mathrm{o}\right)\end{aligned}$$

In this latter, the differentiation of the constant matrix $\mathbf{M}_\mathrm{f}^{(0)}$ w.r.t. $\boldsymbol{\delta}$ yields directly zero, since it is constant. The global mass matrices is now developed according to its non-zero sub-matrices and to the order of the $\boldsymbol{\delta}$-dependency. It yields the following expressions:

$$\begin{aligned}\frac{\partial T}{\partial \boldsymbol{\delta}} &= \frac{1}{2}\,\frac{\partial}{\partial \boldsymbol{\delta}}\left(\mathbf{t}_\mathrm{o}^\top\,\begin{bmatrix}\mathbf{M}_{v\omega}^{(1)}\boldsymbol{\omega}_\mathrm{o}\\ \mathbf{M}_{\omega v}^{(1)}\boldsymbol{v}_\mathrm{o}+\left(\mathbf{M}_{\omega\omega}^{(1)}+\mathbf{M}_{\omega\omega}^{(2)}\right)\boldsymbol{\omega}_\mathrm{o}+\mathbf{M}_{\omega\delta}^{(1)}\dot{\delta}\\ \mathbf{M}_{\delta\omega}^{(1)}\boldsymbol{\omega}_\mathrm{o}\end{bmatrix}\right)\\ &= \frac{1}{2}\,\frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{v}_\mathrm{o}^\top\mathbf{M}_{v\omega}^{(1)}\boldsymbol{\omega}_\mathrm{o}+\boldsymbol{\omega}_\mathrm{o}^\top\mathbf{M}_{\omega v}^{(1)}\boldsymbol{v}_\mathrm{o}+\boldsymbol{\omega}_\mathrm{o}^\top\left(\mathbf{M}_{\omega\omega}^{(1)}+\mathbf{M}_{\omega\omega}^{(2)}\right)\boldsymbol{\omega}_\mathrm{o}+\boldsymbol{\omega}_\mathrm{o}^\top\mathbf{M}_{\omega\delta}^{(1)}\dot{\boldsymbol{\delta}}+\dot{\boldsymbol{\delta}}^\top\mathbf{M}_{\delta\omega}^{(1)}\boldsymbol{\omega}_\mathrm{o}\right)\\ &= \frac{1}{2}\,\frac{\partial}{\partial \boldsymbol{\delta}}\left(2\,\boldsymbol{v}_\mathrm{o}^\top\mathbf{M}_{v\omega}^{(1)}\boldsymbol{\omega}_\mathrm{o}+\boldsymbol{\omega}_\mathrm{o}^\top\left(\mathbf{M}_{\omega\omega}^{(1)}+\mathbf{M}_{\omega\omega}^{(2)}\right)\boldsymbol{\omega}_\mathrm{o}+2\,\boldsymbol{\omega}_\mathrm{o}^\top\mathbf{M}_{\omega\delta}^{(1)}\dot{\boldsymbol{\delta}}\right)\end{aligned}\tag{B.55}$$

Each term is developed separately to obtain the differentiation more easily. The first one reads as follows, reminding that the skew-symmetric matrix satisfies $\mathbf{x}^{\times,\top}==\mathbf{x}^\times$:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{v}_\mathrm{o}^\top\mathbf{M}_{v\omega}^{(1)}\boldsymbol{\omega}_\mathrm{o}\right) &= \frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{v}_\mathrm{o}^\top\,\left(-\mathbf{V}_u\boldsymbol{\delta}\right)^\times\boldsymbol{\omega}_\mathrm{o}\right)\\ &= \frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{v}_\mathrm{o}^\top\,\boldsymbol{\omega}_\mathrm{o}^\times\,\mathbf{V}_u\,\boldsymbol{\delta}\right)\\ &= -\mathbf{V}_u^\top\,\boldsymbol{\omega}_\mathrm{o}^\times\,\boldsymbol{v}_\mathrm{o}\end{aligned}$$

This result yields the sub-matrix $\tilde{\mathbf{M}}_{\delta\mathrm{v}}$ of $\tilde{\mathbf{M}}_\mathrm{f}$ by:

$$\tilde{\mathbf{M}}_{\delta\mathrm{v}} = \left(-\mathbf{M}_{\delta v}\,\boldsymbol{\omega}_\mathrm{o}^\times\right)\,\boldsymbol{v}_\mathrm{o}\tag{B.56}$$

The most cumbersome development is due to the quadratic dependency of the corrective inertias. Their differentiation w.r.t. $\boldsymbol{\delta}$ is obtained as follows:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{\omega}_\mathrm{o}^\top\left(\mathbf{M}_{\omega\omega}^{(1)}+\mathbf{M}_{\omega\omega}^{(2)}\right)\boldsymbol{\omega}_\mathrm{o}\right) &= \frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{\omega}_\mathrm{o}^\top\left(\mathbf{I}_{ru}(\boldsymbol{\delta})+\mathbf{I}_{r\psi}(\boldsymbol{\delta})+\mathbf{I}_{uu}(\boldsymbol{\delta},\boldsymbol{\delta})+\mathbf{I}_{\psi\psi}(\boldsymbol{\delta},\boldsymbol{\delta})\right)\boldsymbol{\omega}_\mathrm{o}\right)\\ &= \frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{\omega}_\mathrm{o}^\top\left(\mathbf{I}_{ru}(\boldsymbol{\delta})+\mathbf{I}_{r\psi}(\boldsymbol{\delta})+\mathbf{I}_{uu}(\boldsymbol{\delta},\boldsymbol{\delta})+\mathbf{I}_{\psi\psi}(\boldsymbol{\delta},\boldsymbol{\delta})\right)\boldsymbol{\omega}_\mathrm{o}\right)\end{aligned}$$

The differentiation is performed for each term separately. The first one yields:

$$\frac{\partial}{\partial \boldsymbol{\delta}} \left(\boldsymbol{\omega}_{\mathrm{o}}^{\top} \mathbf{I}_{ru}(\boldsymbol{\delta}) \boldsymbol{\omega}_{\mathrm{o}} \right) = -\frac{\partial}{\partial \boldsymbol{\delta}} \left(\boldsymbol{\omega}_{\mathrm{o}}^{\top} \langle \mathbf{p}_{\mathrm{a}}, \mathbf{V}_u \boldsymbol{\delta} \rangle \boldsymbol{\omega}_{\mathrm{o}} + \boldsymbol{\omega}_{\mathrm{o}}^{\top} \langle \mathbf{e}_x, \mathbf{W}_u \boldsymbol{\delta} \rangle \boldsymbol{\omega}_{\mathrm{o}} \right)$$

whose the first term is first developed, since the second is similar:

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\delta}} \left(\boldsymbol{\omega}_{\mathrm{o}}^{\top} \langle \mathbf{p}_{\mathrm{a}}, \mathbf{V}_u \boldsymbol{\delta} \rangle \boldsymbol{\omega}_{\mathrm{o}} \right) &= -\frac{\partial}{\partial \boldsymbol{\delta}} \left(\boldsymbol{\omega}_{\mathrm{o}}^{\top} \mathbf{p}_{\mathrm{a}}^{\times} (\mathbf{V}_u \boldsymbol{\delta})^{\times} \boldsymbol{\omega}_{\mathrm{o}} + \boldsymbol{\omega}_{\mathrm{o}}^{\top} (\mathbf{V}_u \boldsymbol{\delta})^{\times} \mathbf{p}_{\mathrm{a}}^{\times} \boldsymbol{\omega}_{\mathrm{o}} \right) \\
&= -\frac{\partial}{\partial \boldsymbol{\delta}} \left(-2 \boldsymbol{\omega}_{\mathrm{o}}^{\top} \mathbf{p}_{\mathrm{a}}^{\times} \boldsymbol{\omega}_{\mathrm{o}}^{\times} \mathbf{V}_u \boldsymbol{\delta} \right) \\
&= 2 \mathbf{V}_u^{\top} \boldsymbol{\omega}_{\mathrm{o}}^{\times} \mathbf{p}_{\mathrm{a}}^{\times} \boldsymbol{\omega}_{\mathrm{o}}
\end{aligned}$$

leading to the following result:

$$-\frac{1}{2} \frac{\partial}{\partial \boldsymbol{\delta}} \left(\boldsymbol{\omega}_{\mathrm{o}}^{\top} \mathbf{I}_{ru}(\boldsymbol{\delta}) \boldsymbol{\omega}_{\mathrm{o}} \right) \triangleq \tilde{\mathbf{I}}_{\mathrm{ru}}(\boldsymbol{\omega}_{\mathrm{o}}) \, \boldsymbol{\omega}_{\mathrm{o}} = -\left(\mathbf{V}_u^{\top} \boldsymbol{\omega}_{\mathrm{o}}^{\times} \mathbf{p}_{\mathrm{a}}^{\times} + \mathbf{W}_u^{\top} \boldsymbol{\omega}_{\mathrm{o}}^{\times} \mathbf{e}_x^{\times} \right) \boldsymbol{\omega}_{\mathrm{o}} \tag{B.57}$$

The second term of the initial summation is given hereunder, using the symmetry property of $\mathbf{J}_S$:

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\delta}} \left(\boldsymbol{\omega}_{\mathrm{o}}^{\top} \mathbf{I}_{r\psi}(\boldsymbol{\delta}) \boldsymbol{\omega}_{\mathrm{o}} \right) &= \frac{\partial}{\partial \boldsymbol{\delta}} \left(\boldsymbol{\omega}_{\mathrm{o}}^{\top} \left(\mathbf{V}_{\psi} \boldsymbol{\delta} \right)^{\times} \mathbf{J}_S \boldsymbol{\omega}_{\mathrm{o}} - \boldsymbol{\omega}_{\mathrm{o}}^{\top} \mathbf{J}_S \left(\mathbf{V}_{\psi} \boldsymbol{\delta} \right)^{\times} \boldsymbol{\omega}_{\mathrm{o}} \right) \\
&= \frac{\partial}{\partial \boldsymbol{\delta}} \left(2 \boldsymbol{\omega}_{\mathrm{o}}^{\top} \left(\mathbf{V}_{\psi} \boldsymbol{\delta} \right)^{\times} \mathbf{J}_S \boldsymbol{\omega}_{\mathrm{o}} \right) \\
&= \frac{\partial}{\partial \boldsymbol{\delta}} \left(-2 \boldsymbol{\delta}^{\top} \mathbf{V}_{\psi}^{\top} \boldsymbol{\omega}_{\mathrm{o}}^{\times} \mathbf{J}_S \boldsymbol{\omega}_{\mathrm{o}} \right) \\
&= -2 \mathbf{V}_{\psi}^{\top} \boldsymbol{\omega}_{\mathrm{o}}^{\times} \mathbf{J}_S \boldsymbol{\omega}_{\mathrm{o}}
\end{aligned}$$

It is denoted as follows in the following:

$$-\frac{1}{2} \frac{\partial}{\partial \boldsymbol{\delta}} \left(\boldsymbol{\omega}_{\mathrm{o}}^{\top} \mathbf{I}_{r\psi}(\boldsymbol{\delta}) \boldsymbol{\omega}_{\mathrm{o}} \right) \triangleq \tilde{\mathbf{I}}_{\mathrm{r}\psi}(\boldsymbol{\omega}_{\mathrm{o}}) \, \boldsymbol{\omega}_{\mathrm{o}} = \mathbf{V}_{\psi}^{\top} \boldsymbol{\omega}_{\mathrm{o}}^{\times} \mathbf{J}_S \boldsymbol{\omega}_{\mathrm{o}} \tag{B.58}$$

The third term is given by more involved computations, as follows:

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\delta}} \left(\boldsymbol{\omega}_{\mathrm{o}}^{\top} \mathbf{I}_{uu}(\boldsymbol{\delta}, \boldsymbol{\delta}) \boldsymbol{\omega}_{\mathrm{o}} \right) &= \frac{\partial}{\partial \boldsymbol{\delta}} \left(\boldsymbol{\omega}_{\mathrm{o}}^{\top} \left(\boldsymbol{\delta}^{\top} \bar{\mathbf{Z}}_{\boldsymbol{uu}} \boldsymbol{\delta} \right) \boldsymbol{\omega}_{\mathrm{o}} - \boldsymbol{\omega}_{\mathrm{o}}^{\top} \left[\boldsymbol{\delta} \right]_{\mathbf{u}}^{\top} \mathbf{Z}_{uu} \left[\boldsymbol{\delta} \right]_{\mathbf{u}} \boldsymbol{\omega}_{\mathrm{o}} \right) \\
&= \frac{\partial}{\partial \boldsymbol{\delta}} \left(\left(\boldsymbol{\delta}^{\top} \bar{\mathbf{Z}}_{\boldsymbol{uu}} \boldsymbol{\delta} \right) \boldsymbol{\omega}_{\mathrm{o}}^{\top} \boldsymbol{\omega}_{\mathrm{o}} - \boldsymbol{\omega}_{\mathrm{o}}^{\top} \left[\boldsymbol{\delta} \right]_{\mathbf{u}}^{\top} \mathbf{Z}_{uu} \left[\boldsymbol{\delta} \right]_{\mathbf{u}} \boldsymbol{\omega}_{\mathrm{o}} \right)
\end{aligned}$$

Its first term is easily differentiated w.r.t. $\boldsymbol{\delta}$, while the second need to compute by hand or

symbolically $\frac{\partial}{\partial \boldsymbol{\delta}}([\boldsymbol{\delta}]_{\mathbf{u}}\,\boldsymbol{\omega}_{\mathrm{o}})$. It yields:

$$\frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{\omega}_{\mathrm{o}}^{\top}\mathbf{I}_{uu}(\boldsymbol{\delta},\boldsymbol{\delta})\boldsymbol{\omega}_{\mathrm{o}}\right) = 2\left(\left(\bar{\mathbf{Z}}_{\boldsymbol{uu}}\boldsymbol{\delta}\right)\,\boldsymbol{\omega}_{\mathrm{o}}^{\top}\boldsymbol{\omega}_{\mathrm{o}} - \mathrm{diag}(\omega_x\,\mathbf{I}_{n_x},\omega_y\,\mathbf{I}_{n_y},\omega_z\mathbf{I}_{n_z},\mathbf{0}_{n_\alpha\times n_\alpha})\,\mathbf{Z}_{uu}\,[\boldsymbol{\delta}]_{\mathbf{u}}\,\boldsymbol{\omega}_{\mathrm{o}}\right)$$

However, by coming back to the integral definition of $\mathbf{I}_{uu}$ in (B.29), a symbolic manipulation allows to re-write this result in an equivalent form, as:

$$-\frac{1}{2}\,\frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{\omega}_{\mathrm{o}}^{\top}\mathbf{I}_{uu}(\boldsymbol{\delta},\boldsymbol{\delta})\boldsymbol{\omega}_{\mathrm{o}}\right) \triangleq \tilde{\mathbf{I}}_{\mathrm{uu}}(\boldsymbol{\delta},\boldsymbol{\omega}_{\mathrm{o}})\,\boldsymbol{\omega}_{\mathrm{o}} = \left([\mathbf{Z}_{uu}\,[\boldsymbol{\delta}]_{\mathbf{u}}\,\boldsymbol{\omega}_{\mathrm{o}}]_{\mathbf{u}} - \bar{\mathbf{Z}}_{\boldsymbol{uu}}\boldsymbol{\delta}\,\boldsymbol{\omega}_{\mathrm{o}}^{\top}\right)\,\boldsymbol{\omega}_{\mathrm{o}} \tag{B.59}$$

The fourth term is obtained with similar computations by coming back to its definition in (B.33):

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{\omega}_{\mathrm{o}}^{\top}\mathbf{I}_{\psi\psi}(\boldsymbol{\delta},\boldsymbol{\delta})\boldsymbol{\omega}_{\mathrm{o}}\right) &= \frac{\partial}{\partial \boldsymbol{\delta}}\left(-\int_{\mathcal{B}}\rho\,\boldsymbol{\omega}_{\mathrm{o}}^{\top}\,\left(\delta\mathbf{p}^{\times}\,(\boldsymbol{\Delta}\,\boldsymbol{\delta})\right)^{\times}\left(\delta\mathbf{p}^{\times}\,(\boldsymbol{\Delta}\,\boldsymbol{\delta})\right)^{\times}\,\boldsymbol{\omega}_{\mathrm{o}}\,\mathrm{d}y\mathrm{d}z\mathrm{d}\overline{s}\right)\\
&= \frac{\partial}{\partial \boldsymbol{\delta}}\left(-\int_{\mathcal{B}}\rho\,\boldsymbol{\omega}_{\mathrm{o}}^{\top}\,\delta\mathbf{u}^{\times}\,\delta\mathbf{u}^{\times}\,\boldsymbol{\omega}_{\mathrm{o}}\,\mathrm{d}y\mathrm{d}z\mathrm{d}\overline{s}\right)\\
&= \frac{\partial}{\partial \boldsymbol{\delta}}\left(-\int_{\mathcal{B}}\rho\,\delta\mathbf{u}^{\top}\,\boldsymbol{\omega}_{\mathrm{o}}\times\,\boldsymbol{\omega}_{\mathrm{o}}^{\times}\,\delta\mathbf{u}\,\mathrm{d}y\mathrm{d}z\mathrm{d}\overline{s}\right)\\
&= -2\int_{\mathcal{B}}\rho\,\boldsymbol{\omega}_{\mathrm{o}}\times\,\boldsymbol{\omega}_{\mathrm{o}}^{\times}\,\delta\mathbf{u}\,\mathrm{d}y\mathrm{d}z\mathrm{d}\overline{s}
\end{aligned}$$

reminding that $\delta\mathbf{u} = (\boldsymbol{\Delta}\,\boldsymbol{\delta})^{\times}\,\delta\mathbf{p}$. The computations can be led symbolically to check that the final results reads as follows:

$$-\frac{1}{2}\,\frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{\omega}_{\mathrm{o}}^{\top}\mathbf{I}_{\psi\psi}(\boldsymbol{\delta},\boldsymbol{\delta})\boldsymbol{\omega}_{\mathrm{o}}\right) \triangleq \tilde{\mathbf{I}}_{\psi\psi}(\boldsymbol{\delta},\boldsymbol{\omega}_{\mathrm{o}})\,\boldsymbol{\omega}_{\mathrm{o}}$$

with

$$\tilde{\mathbf{I}}_{\psi\psi}(\boldsymbol{\delta},\boldsymbol{\omega}_{\mathrm{o}}) = \left(\left([\boldsymbol{\omega}_{\mathrm{o}}]_{\Delta}\,\mathbf{Y}_{\psi\psi}\,[\boldsymbol{\delta}]_{\boldsymbol{\psi}} - \bar{\mathbf{Y}}_{\boldsymbol{\psi\psi}}\boldsymbol{\delta}\,\boldsymbol{\omega}_{\mathrm{o}}^{\top}\right)\mathbf{I}_S - [\mathbf{J}_S\boldsymbol{\omega}_{\mathrm{o}}]_{\Delta}\,\mathbf{Y}_{\psi\psi}\,[\boldsymbol{\delta}]_{\boldsymbol{\psi}}\,\boldsymbol{\omega}_{\mathrm{o}}^{\top}\right) \tag{B.60}$$

where the block-diagonal operator $[.]_{\Delta}$ is defined by:

$$[\mathbf{x}]_{\Delta} = \mathrm{diag}(\mathbf{0}_{n_x\times n_x},\,z\,\mathbf{I}_{n_y},\,y\,\mathbf{I}_{n_z},\,x\,\mathbf{I}_{n_\alpha})$$

for a vector $\mathbf{x} = \begin{bmatrix} x & y & z \end{bmatrix}^{\top} \in \mathbb{R}^3$.

The results in (B.57) to (B.60) are used to summarized the contributions of the corrective

inertias as:

$$-\frac{1}{2}\frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{\omega}_{\mathrm{o}}^{\top}\left(\mathbf{M}_{\omega\omega}^{(1)}+\mathbf{M}_{\omega\omega}^{(2)}\right)\boldsymbol{\omega}_{\mathrm{o}}\right)=\left(\tilde{\mathbf{I}}_{\mathrm{ru}}(\boldsymbol{\omega}_{\mathrm{o}})+\tilde{\mathbf{I}}_{\mathrm{r}\psi}(\boldsymbol{\omega}_{\mathrm{o}})+\tilde{\mathbf{I}}_{\mathrm{uu}}(\boldsymbol{\delta},\boldsymbol{\omega}_{\mathrm{o}})+\tilde{\mathbf{I}}_{\psi\psi}(\boldsymbol{\delta},\boldsymbol{\omega}_{\mathrm{o}})\right)\boldsymbol{\omega}_{\mathrm{o}} \tag{B.61}$$

Eventually, the last terms to differentiate is the flexible contribution to the angular momentum with:

$$\frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{\omega}_{\mathrm{o}}^{\top}\mathbf{M}_{\omega\delta}^{(1)}\dot{\boldsymbol{\delta}}\right)=\frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{\omega}_{\mathrm{o}}^{\top}\left(\mathbf{H}_{uu}(\boldsymbol{\delta})+\mathbf{H}_{\psi\psi}(\boldsymbol{\delta})\right)\boldsymbol{\omega}_{\mathrm{o}}\right)$$

For both terms, it is necessary to come back their definitions. The first one is developed using (B.31):

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{\omega}_{\mathrm{o}}^{\top}\mathbf{H}_{uu}(\boldsymbol{\delta})\dot{\boldsymbol{\delta}}\right)&=\frac{\partial}{\partial \boldsymbol{\delta}}\left(\int_0^l \rho S\,\boldsymbol{\omega}_{\mathrm{o}}^{\top}\,(\boldsymbol{\Phi}\,\boldsymbol{\delta})^{\times}\,\boldsymbol{\Phi}\,\dot{\boldsymbol{\delta}}\mathrm{d}\bar{s}\right)\\&=\frac{\partial}{\partial \boldsymbol{\delta}}\left(-\int_0^l \rho S\,\boldsymbol{\delta}^{\top}\,\boldsymbol{\Phi}^{\top}\,\boldsymbol{\omega}_{\mathrm{o}}^{\times}\,\boldsymbol{\Phi}\,\dot{\boldsymbol{\delta}}\mathrm{d}\bar{s}\right)\\&=-\int_0^l \rho S\,\boldsymbol{\Phi}^{\top}\,\boldsymbol{\omega}_{\mathrm{o}}^{\times}\,\boldsymbol{\Phi}\,\dot{\boldsymbol{\delta}}\mathrm{d}\bar{s}\end{aligned}$$

Again, either by hand or symbolically, one obtains the following results:

$$-\frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{\omega}_{\mathrm{o}}^{\top}\mathbf{H}_{uu}(\boldsymbol{\delta})\dot{\boldsymbol{\delta}}\right)\triangleq\tilde{\mathbf{H}}_{\mathrm{uu}}(\boldsymbol{\omega}_{\mathrm{o}})\dot{\boldsymbol{\delta}}=\left[\begin{array}{ccc|c}\mathbf{0}&-\omega_z\mathbf{Z}_{xy}&\omega_y\mathbf{Z}_{xz}&\mathbf{0}\\\omega_z\mathbf{Z}_{yx}&\mathbf{0}&-\omega_x\mathbf{Z}_{yz}&\mathbf{0}\\-\omega_y\mathbf{Z}_{zx}&\omega_x\mathbf{Z}_{zy}&\mathbf{0}&\mathbf{0}\\\hline\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\end{array}\right]\dot{\boldsymbol{\delta}} \tag{B.62}$$

For the second one, the computations based on (B.32) and (B.35) lead to:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{\omega}_{\mathrm{o}}^{\top}\mathbf{H}_{\psi\psi}(\boldsymbol{\delta})\dot{\boldsymbol{\delta}}\right)&=\frac{\partial}{\partial \boldsymbol{\delta}}\left(\int_{\mathcal{B}}\rho\,\boldsymbol{\omega}_{\mathrm{o}}^{\top}\,(\delta\mathbf{p}\times\boldsymbol{\Delta}\boldsymbol{\delta})^{\times}\,\delta\mathbf{p}^{\times}\boldsymbol{\Delta}\dot{\boldsymbol{\delta}}\,\mathrm{d}y\mathrm{d}z\mathrm{d}\bar{s}\right)\\&=\frac{\partial}{\partial \boldsymbol{\delta}}\left(-\int_{\mathcal{B}}\rho\,(\delta\mathbf{p}\times\boldsymbol{\Delta}\boldsymbol{\delta})^{\top}\,\boldsymbol{\omega}_{\mathrm{o}}^{\times}\delta\mathbf{p}^{\times}\boldsymbol{\Delta}\dot{\boldsymbol{\delta}}\,\mathrm{d}y\mathrm{d}z\mathrm{d}\bar{s}\right)\\&=\frac{\partial}{\partial \boldsymbol{\delta}}\left(-\int_{\mathcal{B}}\rho\,\boldsymbol{\delta}^{\top}\,\boldsymbol{\Delta}^{\top}\,\delta\mathbf{p}^{\times,\top}\,\boldsymbol{\omega}_{\mathrm{o}}^{\times}\delta\mathbf{p}^{\times}\boldsymbol{\Delta}\dot{\boldsymbol{\delta}}\,\mathrm{d}y\mathrm{d}z\mathrm{d}\bar{s}\right)\\&=\frac{\partial}{\partial \boldsymbol{\delta}}\left(\int_{\mathcal{B}}\rho\,\boldsymbol{\delta}^{\top}\,\boldsymbol{\Delta}^{\top}\,\delta\mathbf{p}^{\times}\,\boldsymbol{\omega}_{\mathrm{o}}^{\times}\delta\mathbf{p}^{\times}\boldsymbol{\Delta}\dot{\boldsymbol{\delta}}\,\mathrm{d}y\mathrm{d}z\mathrm{d}\bar{s}\right)\\&=\int_{\mathcal{B}}\rho\,\boldsymbol{\Delta}^{\top}\,\delta\mathbf{p}^{\times}\,\boldsymbol{\omega}_{\mathrm{o}}^{\times}\delta\mathbf{p}^{\times}\boldsymbol{\Delta}\dot{\boldsymbol{\delta}}\,\mathrm{d}y\mathrm{d}z\mathrm{d}\bar{s}\end{aligned}$$

which yields by hand or symbolically:

$$-\frac{\partial}{\partial \boldsymbol{\delta}}\left(\boldsymbol{\omega}_{\mathrm{o}}^{\top}\mathbf{H}_{\psi\psi}(\boldsymbol{\delta})\dot{\boldsymbol{\delta}}\right) \triangleq \tilde{\mathbf{H}}_{\psi\psi}(\boldsymbol{\omega}_{\mathrm{o}})\,\dot{\boldsymbol{\delta}} = \left[\begin{array}{c|ccc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \tilde{\omega}_y\mathbf{Y}_{y\alpha} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\tilde{\omega}_z\mathbf{Y}_{z\alpha} \\ \mathbf{0} & -\tilde{\omega}_y\mathbf{Y}_{\alpha y} & \tilde{\omega}_z\mathbf{Y}_{\alpha z} & \mathbf{0} \end{array}\right]\dot{\boldsymbol{\delta}} \tag{B.63}$$

where the angular rate is weighted by the moment of area with $\tilde{\boldsymbol{\omega}}_o = \mathbf{I}_S\boldsymbol{\omega}_{\mathrm{o}}$.

Back to (B.55), the general expression of the state derivatives is set under matrix form to define $\tilde{\mathbf{M}}_{\mathrm{f}}$ of (B.49) as follows:

$$\tilde{\mathbf{M}}_{\mathrm{f}} = \left[\begin{array}{ccc|c} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n_f} \\ \tilde{\mathbf{M}}_{\omega\mathrm{v}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times n_f} \\ \hline \tilde{\mathbf{M}}_{\delta\mathrm{v}} & \tilde{\mathbf{M}}_{\delta\omega} & \tilde{\mathbf{M}}_{\delta\delta} \end{array}\right] \tag{B.64}$$

with the sub-matrices given by:

$$\begin{aligned} \tilde{\mathbf{M}}_{\omega\mathrm{v}} &= -\left(\mathring{\mathbf{M}}_{\omega v} + \boldsymbol{\omega}_{\mathrm{o}}^{\times}\mathbf{M}_{\omega v} - \mathbf{M}_{\omega v}\boldsymbol{\omega}_{\mathrm{o}}^{\times}\right) \\ \tilde{\mathbf{M}}_{\delta\mathrm{v}} &= -\left(-\mathbf{M}_{\delta v}\boldsymbol{\omega}_{\mathrm{o}}^{\times}\right) \\ \tilde{\mathbf{M}}_{\delta\omega} &= \tilde{\mathbf{I}}_{\mathrm{ru}} + \tilde{\mathbf{I}}_{\mathrm{r}\psi} + \tilde{\mathbf{I}}_{\mathrm{uu}} + \tilde{\mathbf{I}}_{\psi\psi} \\ \tilde{\mathbf{M}}_{\delta\delta} &= \tilde{\mathbf{H}}_{\mathrm{uu}} + \tilde{\mathbf{H}}_{\psi\psi} \end{aligned}$$

The corrective terms of size $\mathbb{R}^{n_f\times3}$ on the second moment of inertia are recalled here as a reminder from (B.57) to (B.60) :

$$\begin{aligned} \tilde{\mathbf{I}}_{\mathrm{ru}}(\boldsymbol{\omega}_{\mathrm{o}}) &= -\mathbf{V}_u^{\top}\,\boldsymbol{\omega}_{\mathrm{o}}^{\times}\,\mathbf{p}_a^{\times} - \mathbf{W}_u^{\top}\,\boldsymbol{\omega}_{\mathrm{o}}^{\times}\,\mathbf{e}_x^{\times} \\ \tilde{\mathbf{I}}_{\mathrm{r}\psi}(\boldsymbol{\omega}_{\mathrm{o}}) &= \mathbf{V}_{\psi}^{\top}\,\boldsymbol{\omega}_{\mathrm{o}}^{\times}\,\mathbf{J}_S \\ \tilde{\mathbf{I}}_{\mathrm{uu}}(\boldsymbol{\delta},\boldsymbol{\omega}_{\mathrm{o}}) &= \left[\mathbf{Z}_{uu}\left[\boldsymbol{\delta}\right]_{\mathbf{u}}\boldsymbol{\omega}_{\mathrm{o}}\right]_{\mathbf{u}} - \bar{\mathbf{Z}}_{\boldsymbol{uu}}\boldsymbol{\delta}\,\boldsymbol{\omega}_{\mathrm{o}}^{\top} \\ \tilde{\mathbf{I}}_{\psi\psi}(\boldsymbol{\delta},\boldsymbol{\omega}_{\mathrm{o}}) &= \left(\left[\boldsymbol{\omega}_{\mathrm{o}}\right]_{\Delta}\mathbf{Y}_{\psi\psi}\left[\boldsymbol{\delta}\right]_{\psi} - \bar{\mathbf{Y}}_{\psi\boldsymbol{\psi}\boldsymbol{\delta}}\,\boldsymbol{\omega}_{\mathrm{o}}^{\top}\right)\mathbf{I}_S - \left[\mathbf{J}_S\boldsymbol{\omega}_{\mathrm{o}}\right]_{\Delta}\mathbf{Y}_{\psi\psi}\left[\boldsymbol{\delta}\right]_{\psi}\boldsymbol{\omega}_{\mathrm{o}}^{\top} \end{aligned}$$

The corrective terms of size $\mathbb{R}^{n_f \times n_f}$ on the angular momentum are given in (B.62) and (B.63) by:

$$\tilde{\mathbf{H}}_{\mathrm{uu}}(\boldsymbol{\omega}_{\mathrm{o}}) = \left[\begin{array}{ccc|c} \mathbf{0} & -\omega_z \mathbf{Z}_{xy} & \omega_y \mathbf{Z}_{xz} & \mathbf{0} \\ \omega_z \mathbf{Z}_{yx} & \mathbf{0} & -\omega_x \mathbf{Z}_{yz} & \mathbf{0} \\ -\omega_y \mathbf{Z}_{zx} & \omega_x \mathbf{Z}_{zy} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right]$$

$$\tilde{\mathbf{H}}_{\psi\psi}(\boldsymbol{\omega}_{\mathrm{o}}) = \left[\begin{array}{c|ccc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \tilde{\omega}_y \mathbf{Y}_{y\alpha} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\tilde{\omega}_z \mathbf{Y}_{z\alpha} \\ \mathbf{0} & -\tilde{\omega}_y \mathbf{Y}_{\alpha y} & \tilde{\omega}_z \mathbf{Y}_{\alpha z} & \mathbf{0} \end{array}\right]$$

APPENDIX C SPACE ROBOT MODELING DETAILS

This appendix gathers all the results and computations used in the modeling of a space robot and the corresponding algorithms. These objects are considered as a rigid hub supporting many appendages, either rigid or flexible. The system is illustrated in Figure C.1 with all the frames and index notations.

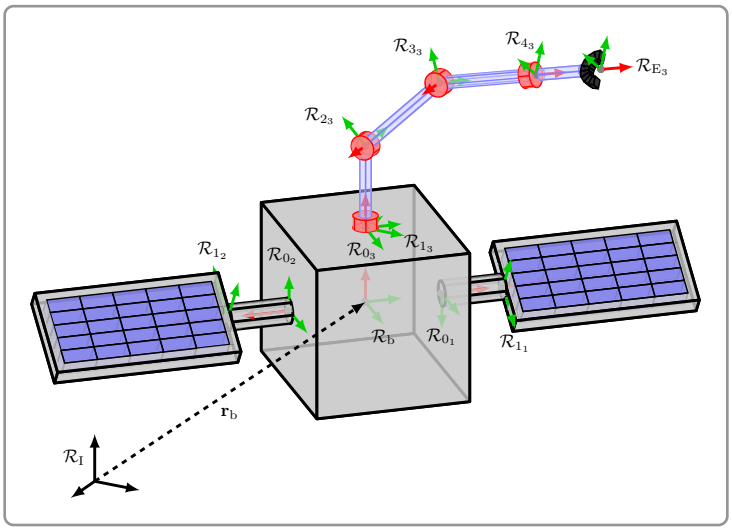


Figure C.1: Frames assignment on a spacecraft with various appendages

C.1 Space Robot Dynamics Algorithms

This section gathers the dynamics algorithms dedicated to space robots. They are provided to compute recursively the mass matrix, the vector of Coriolis and centrifugal forces, and the forward dynamics using the DeNOC approach.

C.1.1 Mass Matrix Algorithm

Similar to the algorithms presented for robotic systems, the next algorithm computes the mass matrix of a space robot assuming a "star shape" with all the chain-like appendages connected to a rigid hub, as shown in Figure C.1. The approach is still based on the *Composite Rigid Body* concept to describe each appendage (Featherstone, 2008). The global inertia of the composite system is obtained by merging all their own inertias with the spacecraft one, expressed at its CoM G_b. The proposed algorithm starts with the DeNOC formalism used to describe tree-type systems in (Shah et al., 2012a) and then derives explicitly the algorithm by computations similar to the fixed-base robotics case.

Reminder: The mass matrix is developed explicitly in (6.52) to emphasize how the dynamics of the appendages interacts with the spacecraft:

$$\begin{aligned}\mathbf{D} &= \mathbf{N}^{\top}\,[\mathbf{M}]\,\mathbf{N} \\ &= \left[\begin{array}{c|cccc} \mathbf{D}_{\mathrm{bb}} & \mathbf{D}_{\mathrm{ba}_1} & \dots & \dots & \mathbf{D}_{\mathrm{ba}_{n_\mathrm{a}}} \\ \hline * & \mathbf{D}_{\mathrm{a}_1} & \mathbf{0} & \dots & \mathbf{0} \\ * & \mathbf{0} & \ddots & \ddots & \vdots \\ * & \vdots & \ddots & \ddots & \mathbf{0} \\ * & \mathbf{0} & \dots & \mathbf{0} & \mathbf{D}_{\mathrm{a}_{n_\mathrm{a}}} \end{array}\right]\end{aligned} \tag{C.1}$$

where the sub-blocks are given in (6.53) by:

$$\mathbf{D}_{\mathrm{bb}} = {}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}}^{\top}\mathbf{M}_{\mathrm{b}}\,{}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}} + \textstyle\sum_k {}^{k}\mathbf{N}_{\mathrm{b}}^{\top}\,[\mathbf{M}_{\mathrm{a}_k}]\,{}^{k}\mathbf{N}_{\mathrm{b}} \tag{C.2a}$$

$$\mathbf{D}_{\mathrm{a}_k} = {}^{k}\mathbf{N}_{k}^{\top}\,[\mathbf{M}_{\mathrm{a}_k}]\,{}^{k}\mathbf{N}_{k} \tag{C.2b}$$

$$\mathbf{D}_{\mathrm{ba}_k} = {}^{k}\mathbf{N}_{\mathrm{b}}^{\top}\,[\mathbf{M}_{\mathrm{a}_k}]\,{}^{k}\mathbf{N}_{k} \tag{C.2c}$$

Theirs expressions lie on the DeNOC matrix $\mathbf{N}$ according to (6.27):

$$\begin{bmatrix} \mathbf{t}_{\mathrm{b}} \\ \mathbf{t}_{\mathrm{a}_1} \\ \mathbf{t}_{\mathrm{a}_2} \\ \vdots \\ \mathbf{t}_{\mathrm{a}_{n_\mathrm{a}}} \end{bmatrix} = \underbrace{\begin{bmatrix} {}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ {}^{1}\mathbf{N}_{\mathrm{b}} & {}^{1}\mathbf{N}_{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ {}^{2}\mathbf{N}_{\mathrm{b}} & \mathbf{0} & {}^{2}\mathbf{N}_{2} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ {}^{n_\mathrm{a}}\mathbf{N}_{\mathrm{b}} & \mathbf{0} & \dots & \mathbf{0} & {}^{n_\mathrm{a}}\mathbf{N}_{n_\mathrm{a}} \end{bmatrix}}_{=\mathbf{N}} \begin{bmatrix} \dot{\mathbf{q}}_{\mathrm{b}} \\ \dot{\mathbf{q}}_{\mathrm{a}_1} \\ \dot{\mathbf{q}}_{\mathrm{a}_2} \\ \vdots \\ \dot{\mathbf{q}}_{\mathrm{a}_{n_\mathrm{a}}} \end{bmatrix} \tag{C.3}$$

whose decoupled form is detailed in (6.29) as:

$$\mathbf{N} = \underbrace{\begin{bmatrix} \mathbf{I}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ {}^{1}\mathbf{N}_{\mathrm{b},l} & {}^{1}\mathbf{N}_{1,l} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ {}^{2}\mathbf{N}_{\mathrm{b},l} & \mathbf{0} & {}^{2}\mathbf{N}_{2,l} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ {}^{n_\mathrm{a}}\mathbf{N}_{\mathrm{b},l} & \mathbf{0} & \dots & \mathbf{0} & {}^{n_\mathrm{a}}\mathbf{N}_{n_\mathrm{a},l} \end{bmatrix}}_{=\mathbf{N}_l} \underbrace{\begin{bmatrix} {}^{\mathrm{b}}\mathbf{N}_{d} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^{1}\mathbf{N}_{d} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & {}^{2}\mathbf{N}_{d} & \ddots & \vdots \\ \mathbf{0} & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & {}^{n_\mathrm{a}}\mathbf{N}_{d} \end{bmatrix}}_{=\mathbf{N}_d} \tag{C.4}$$

Each of these elements is actually part of the decoupled DeNOC matrix obtained for one appendage with a fixed base in appendix C.1. Their main properties are recalled here for the next developments:

$$
\begin{aligned}
{}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}} &= {}^{\mathrm{b}}\mathbf{N}_{\mathrm{b},l}\,{}^{\mathrm{b}}\mathbf{N}_{d} = \mathbf{I}_{6}\,\mathbf{P}_{\mathrm{b}} && \text{Decoupled form at the spacecraft level}\\
{}^{k}\mathbf{N}_{k} &= {}^{k}\mathbf{N}_{k,l}\,{}^{k}\mathbf{N}_{d} && \text{Decoupled form at the appendage level}\\
{}^{k}\mathbf{N}_{\mathrm{b}} &= {}^{k}\mathbf{N}_{\mathrm{b},l}\,{}^{\mathrm{b}}\mathbf{N}_{d} && \text{Decoupled form of the coupling terms}\\
\mathbf{D}_{\mathrm{a}_k} &= {}^{k}\mathbf{N}_{k}^{\top}\,[\mathbf{M}_{\mathrm{a}_k}]\,{}^{k}\mathbf{N}_{k} && \text{Mass matrix decomposition of an appendage}\\
\widehat{\mathbf{M}}_{\mathrm{a}_k} &= {}^{k}\mathbf{N}_{k,l}^{\top}\,[\mathbf{M}_{\mathrm{a}_k}]\,{}^{k}\mathbf{N}_{k,l} && \text{Composite body inertias of an appendage}
\end{aligned}
$$

where the structure of the matrix ${}^{k}\mathbf{N}_{\mathrm{b},l}$ is also recalled for sake of completeness by (6.21):

$$
{}^{k}\mathbf{N}_{\mathrm{b},l} = \begin{bmatrix} \mathbf{A}_{1_k,\mathrm{b}} \\ \mathbf{A}_{2_k,\mathrm{b}} \\ \vdots \\ \mathbf{A}_{n_k,\mathrm{b}} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{2_k,1_k} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_{3_k,1_k} & \mathbf{A}_{3_k,2_k} & \mathbf{I} & \ddots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{A}_{n_k,1_k} & \mathbf{A}_{n_k,2_k} & \cdots & \mathbf{A}_{n_k,n_k-1} & \mathbf{I} \end{bmatrix}}_{={}^{k}\mathbf{N}_{k,l}} \underbrace{\begin{bmatrix} \mathbf{A}_{1_k,\mathrm{b}} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}}_{={}^{k}\mathbf{A}_{\mathrm{b}}} \tag{C.5}
$$

In the first place, the diagonal terms of $\mathbf{D}$ describing the appendages self-dynamics are obtained by the fixed-base algorithm in Algorithm A.1.

Secondly, the coupling terms $\mathbf{D}_{\mathrm{ba}_k}$ are developed as follows:

$$
\begin{aligned}
\mathbf{D}_{\mathrm{ba}_k} &= {}^{k}\mathbf{N}_{\mathrm{b}}^{\top}\,[\mathbf{M}_{\mathrm{a}_k}]\,{}^{k}\mathbf{N}_{k}\\
&= {}^{\mathrm{b}}\mathbf{N}_{d}^{\top}\,{}^{k}\mathbf{N}_{\mathrm{b},l}^{\top}\,[\mathbf{M}_{\mathrm{a}_k}]\,{}^{k}\mathbf{N}_{k,l}\,{}^{k}\mathbf{N}_{d}\\
&= {}^{\mathrm{b}}\mathbf{N}_{d}^{\top}\,{}^{k}\mathbf{A}_{\mathrm{b}}^{\top}\,{}^{k}\mathbf{N}_{k,l}^{\top}\,[\mathbf{M}_{\mathrm{a}_k}]\,{}^{k}\mathbf{N}_{k,l}\,{}^{k}\mathbf{N}_{d}\\
&= {}^{\mathrm{b}}\mathbf{N}_{d}^{\top}\,{}^{k}\mathbf{A}_{\mathrm{b}}^{\top}\,\widehat{\mathbf{M}}_{\mathrm{a}_k}\,{}^{k}\mathbf{N}_{d}\\
&= \mathbf{P}_{\mathrm{b}}^{\top}\,\mathbf{A}_{1_k,\mathrm{b}}^{\top}\left[\;\widehat{\mathbf{M}}_{1_k}\mathbf{P}_{1_k}\;\middle|\;\mathbf{A}_{2_k,1_k}^{\top}\widehat{\mathbf{M}}_{2_k}\mathbf{P}_{2_k}\;\middle|\;\ldots\;\middle|\;\mathbf{A}_{n_k,n_k-1}^{\top}\widehat{\mathbf{M}}_{n_k}\mathbf{P}_{n_k}\;\right]
\end{aligned}
$$

where the matrix on the RHS is already computed in the algorithm for fixed-base manipulators. The coupling terms are thus described by row block-structures, whose elements are denoted by:

$$
\mathbf{D}_{\mathrm{ba}_k}(i_k) = \mathbf{P}_{\mathrm{b}}^{\top}\,\mathbf{A}_{1_k,\mathrm{b}}^{\top}\,\mathbf{A}_{i_k,1_k}^{\top}\widehat{\mathbf{M}}_{i_k}\mathbf{P}_{i_k}
$$

Eventually, the spacecraft term is the most cumbersome one since all the appendage inertias are merged at this step:

$$
\begin{aligned}
\mathbf{D}_{\mathrm{bb}} &= {}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}}^{\top}\mathbf{M}_{\mathrm{b}}\,{}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}} + \textstyle\sum_k {}^{k}\mathbf{N}_{\mathrm{b}}^{\top}\,[\mathbf{M}_{\mathrm{a}_k}]\,{}^{k}\mathbf{N}_{\mathrm{b}} \\
&= {}^{\mathrm{b}}\mathbf{N}_{d}^{\top}\mathbf{M}_{\mathrm{b}}\,{}^{\mathrm{b}}\mathbf{N}_{d} + \textstyle\sum_k {}^{\mathrm{b}}\mathbf{N}_{d}^{\top}\,{}^{k}\mathbf{N}_{b,l}^{\top}\,[\mathbf{M}_{\mathrm{a}_k}]\,{}^{k}\mathbf{N}_{b,l}\,{}^{\mathrm{b}}\mathbf{N}_{d} \\
&= {}^{\mathrm{b}}\mathbf{N}_{d}^{\top}\left(\mathbf{M}_{\mathrm{b}} + \textstyle\sum_k {}^{k}\mathbf{A}_{\mathrm{b}}^{\top}\,{}^{k}\mathbf{N}_{k,l}^{\top}\,[\mathbf{M}_{\mathrm{a}_k}]\,{}^{k}\mathbf{N}_{k,l}\,{}^{k}\mathbf{A}_{\mathrm{b}}\right){}^{\mathrm{b}}\mathbf{N}_{d} \\
&= \mathbf{P}_{\mathrm{b}}^{\top}\left(\mathbf{M}_{\mathrm{b}} + \textstyle\sum_k {}^{k}\mathbf{A}_{\mathrm{b}}^{\top}\,\widehat{\mathbf{M}}_{\mathrm{a}_k}\,{}^{k}\mathbf{A}_{\mathrm{b}}\right)\mathbf{P}_{\mathrm{b}} \\
&= \mathbf{P}_{\mathrm{b}}^{\top}\left(\mathbf{M}_{\mathrm{b}} + \textstyle\sum_k \mathbf{A}_{1_k,\mathrm{b}}^{\top}\,\widehat{\mathbf{M}}_{1_k}\,\mathbf{A}_{1_k,\mathrm{b}}\right)\mathbf{P}_{\mathrm{b}}
\end{aligned}
$$

The previous relations are merged to derive the global algorithm of computation of $\mathbf{D}$. Similarly to the fixed-base case, a first inward loop over each appendage computes their own mass matrix and their coupling terms. Then, the spacecraft mass matrix is obtained by gathering the appendage inertias.

At the appendage level, $\forall k = 1 \dots n_{\mathrm{a}}$,

$$\widehat{\mathbf{M}}_{n_k+1} = \mathbf{M}_{n_k+1} \quad \text{(if payload)} \tag{C.6a}$$

$\forall i_k = n_k \dots 1_k$,

$$\widehat{\mathbf{M}}_{i_k} = \mathbf{M}_{i_k} + \mathbf{A}_{i_k+1,i_k}^{\top}\,\widehat{\mathbf{M}}_{i_k+1}\,\mathbf{A}_{i_k+1,i_k} \tag{C.6b}$$

$\forall j_k \leq i_k$,

$$\mathbf{D}_{\mathrm{a}_k}(i_k, j_k) = \mathbf{P}_{i_k}^{\top}\,\widehat{\mathbf{M}}_{i_k}\,\mathbf{A}_{i_k,j_k}\,\mathbf{P}_{j_k} \tag{C.6c}$$

$\forall j_k > i_k$,

$$\mathbf{D}_{\mathrm{a}_k}(i_k, j_k) = \mathbf{D}(j_k, i_k)^{\top} \tag{C.6d}$$

$$\mathbf{D}_{\mathrm{ba}_k}(i_k) = \mathbf{P}_{\mathrm{b}}^{\top}\,\mathbf{A}_{1_k,\mathrm{b}}^{\top}\,\mathbf{A}_{i_k,1_k}^{\top}\,\widehat{\mathbf{M}}_{i_k}\,\mathbf{P}_{i_k} \tag{C.6e}$$

At the spacecraft level,

$$\mathbf{D}_{\mathrm{bb}} = \mathbf{P}_{\mathrm{b}}^{\top}\left(\mathbf{M}_{\mathrm{b}} + \textstyle\sum_k \mathbf{A}_{1_k,\mathrm{b}}^{\top}\,\widehat{\mathbf{M}}_{1_k}\,\mathbf{A}_{1_k,\mathrm{b}}\right)\mathbf{P}_{\mathrm{b}} \tag{C.6f}$$

Algorithm C.1: Recursive computation of the mass matrix of a space robot

Function : $\mathbf{D} = \texttt{MassMatSpace}(\mathbf{q})$

Input : $\mathbf{q}$

Output : $\mathbf{D}$

Data : $\left\{ {}^{i_k-1}R^{\mathrm{r/f}}_{i_k}\,,\ \mathbf{A}^{\mathrm{r/f}}_{i_k,i_k-1}\,,\ \mathbf{P}_{i_k}\,,\ \mathbf{M}_{i_k}\,,\ \mathbf{A}_{0_k,\mathrm{b}}\,,\ \mathbf{M}_{\mathrm{b}}\,,\ \mathbf{P}_{\mathrm{b}}\ ;\ i_k = 0_k \dots n_k + 1,\ k = 1 \dots n_{\mathrm{a}} \right\}$

$\texttt{Initialize}\ \left(\widehat{\mathbf{M}}_{\mathrm{a}} = \mathbf{0}_{6\times 6}\right)$ // **APPENDAGE TERMS $\mathbf{D}_{\mathrm{a}_K}$ AND $\mathbf{D}_{\mathrm{ba}_K}$**

for $k = 1 \dots n_{\mathrm{a}}$ **do**

  $\texttt{Initialize}\ \left(\widehat{\mathbf{M}}_{n_k+1} = \mathbf{M}_{n_k+1} = \mathbf{M}_{payload,k}\right)$

  for $i = n_k \dots 1_k$ **do** // Recursive computation of $\widehat{\mathbf{M}}_{i_k}$

    ${}^{i_k+1}R_{i_k} = {}^{i_k}R^{\mathrm{r}}_{i_k+1}(q_{i_k+1})^{\top}$ or ${}^{i_k}R^{\mathrm{f}}_{i_k+1}(q_{i_k+1}, \boldsymbol{\delta}_{i_k})^{\top}$ (4.22) and (5.35)

    ${}^{i_k+1}\boldsymbol{R}_{i_k} = \mathrm{diag}({}^{i_k+1}R_{i_k}, {}^{i_k+1}R_{i_k})$ (4.31)

    $\mathbf{A}_{i_k+1,i_k} = {}^{i_k+1}\boldsymbol{R}_{i_k}\ \mathbf{A}^{\mathrm{r/f}}_{i_k+1,i_k}$ (4.30a) and (5.45a)

    $\widehat{\mathbf{M}}_{i_k} = \mathbf{M}_{i_k} + \mathbf{A}^{\top}_{i_k+1,i_k}\ \widehat{\mathbf{M}}_{i_k+1}\ \mathbf{A}_{i_k+1,i_k}$ (C.6b)

    $\mathbf{A}_{i_k,i_k} = \mathbf{I}_6$

    for $j_k = i_k \dots 1_k$ **do** // Computation of $\mathbf{D}_{\mathrm{a}_k}$

      $\mathbf{D}_{\mathrm{a}_k}(i_k, j_k) = \mathbf{P}^{\top}_{i_k}\ \widehat{\mathbf{M}}_{i_k}\ \mathbf{A}_{i_k,j_k}\ \mathbf{P}_{j_k}$ (C.6c)

      $\mathbf{D}_{\mathrm{a}_k}(j_k, i_k) = \mathbf{D}(i_k, j_k)^{\top}$ (C.6d)

      ${}^{j_k}R_{j_k-1} = {}^{j_k-1}R^{\mathrm{r}}_{j_k}(q_{j_k})^{\top}$ or ${}^{j_k-1}R^{\mathrm{f}}_{j_k}(q_{j_k}, \boldsymbol{\delta}_{j_k-1})^{\top}$ // Update of the $\mathbf{A}_{i_k,j_k}$

      ${}^{j_k}\boldsymbol{R}_{j_k-1} = \mathrm{diag}({}^{j_k}R_{j_k-1}, {}^{j_k}R_{j_k-1})$

      $\mathbf{A}_{i_k,j_k-1} = \mathbf{A}_{i_k,j_k}\ \left({}^{j_k}\boldsymbol{R}_{j_k-1}\ \mathbf{A}^{\mathrm{r/f}}_{j_k,j_k-1}\right)$ // Computation of $\mathbf{D}_{\mathrm{ba}_k}$

    $\mathbf{D}_{\mathrm{ba}_k}(i_k) = \mathbf{P}^{\top}_{\mathrm{b}}\ \mathbf{A}^{\top}_{0_k,\mathrm{b}}\ \mathbf{A}^{\top}_{i_k,0_k}\ \widehat{\mathbf{M}}_{i_k}\ \mathbf{P}_{i_k}$ (C.6e)

  $\mathbf{A}_{1_k,\mathrm{b}} = \mathbf{A}_{1_k,0_k}, \mathbf{A}_{0_k,\mathrm{b}}$ // Update of $\widehat{\mathbf{M}}_{\mathrm{a}}$

  $\widehat{\mathbf{M}}_{\mathrm{a}} = \widehat{\mathbf{M}}_{\mathrm{a}} + \mathbf{A}^{\top}_{1_k,\mathrm{b}}\ \widehat{\mathbf{M}}_{1_k}\ \mathbf{A}_{1_k,\mathrm{b}}$ (C.6f)

// **SPACECRAFT TERM $\mathbf{D}_{\mathrm{bb}}$**

$\mathbf{D}_{\mathrm{bb}} = \mathbf{P}^{\top}_{\mathrm{b}}\ \left(\mathbf{M}_{\mathrm{b}} + \widehat{\mathbf{M}}_{\mathrm{a}}\right)\mathbf{P}_{\mathrm{b}}$ (C.6f)

---

return D

C.1.2 Coriolis and Centrifugal Matrix Algorithm

Using the previous developments for the chain-like manipulators, the Coriolis and centrifugal matrix $\mathbf{C}$ and vector $\mathbf{h}$ are derived by recursive algorithms. Their analytic expressions are recalled from the thesis content, and the detailed expressions of the DeNOC matrices $\mathbf{N}_l$ and $\mathbf{N}_d$ are already given above in (C.3) and (C.4).

Reminder:

As for the mass matrix, the matrix of nonlinear efforts $\mathbf{C}$ reflects the dynamic structure of the space robot. From (6.54), it is recalled that:

$$\begin{aligned}\mathbf{C} &= \mathbf{N}^\top \left[\mathbf{M}\right] \dot{\mathbf{N}} + \mathbf{N}^\top \left[\dot{\mathbf{M}}\right] \mathbf{N} \\ &= \left[\begin{array}{c|cccc} \mathbf{C}_{\mathrm{bb}} & \mathbf{C}_{\mathrm{ba}_1} & \dots & \dots & \mathbf{C}_{\mathrm{ba}_{n_\mathrm{a}}} \\ \hline \mathbf{C}_{\mathrm{a}_1\mathrm{b}} & \mathbf{C}_{\mathrm{a}_1} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \mathbf{0} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{C}_{\mathrm{a}_{n_\mathrm{a}}\mathrm{b}} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{C}_{\mathrm{a}_{n_\mathrm{a}}} \end{array}\right]\end{aligned} \tag{C.7}$$

with the sub-matrices given by (6.53):

$$\mathbf{C}_{\mathrm{bb}} = {}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}}^\top \left(\mathbf{M}_{\mathrm{b}}\, {}^{\mathrm{b}}\dot{\mathbf{N}}_{\mathrm{b}} + \dot{\mathbf{M}}_{\mathrm{b}}\, {}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}}\right) + \textstyle\sum_k {}^{k}\mathbf{N}_{\mathrm{b}}^\top \left(\left[\mathbf{M}_{\mathrm{a}_k}\right] {}^{k}\dot{\mathbf{N}}_{\mathrm{b}} + \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right] {}^{k}\mathbf{N}_{\mathrm{b}}\right) \tag{C.8a}$$

$$\mathbf{C}_{\mathrm{ba}_k} = {}^{k}\mathbf{N}_{\mathrm{b}}^\top \left(\left[\mathbf{M}_{\mathrm{a}_k}\right] {}^{k}\dot{\mathbf{N}}_{k} + \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right] {}^{k}\mathbf{N}_{k}\right) \tag{C.8b}$$

$$\mathbf{C}_{\mathrm{a}_k\mathrm{b}} = {}^{k}\mathbf{N}_{k}^\top \left(\left[\mathbf{M}_{\mathrm{a}_k}\right] {}^{k}\dot{\mathbf{N}}_{\mathrm{b}} + \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right] {}^{k}\mathbf{N}_{\mathrm{b}}\right) \tag{C.8c}$$

$$\mathbf{C}_{\mathrm{a}_k} = {}^{k}\mathbf{N}_{k}^\top \left(\left[\mathbf{M}_{\mathrm{a}_k}\right] {}^{k}\dot{\mathbf{N}}_{k} + \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right] {}^{k}\mathbf{N}_{k}\right) \tag{C.8d}$$

The intermediate variables used in appendix A.3.3 to compute the fixed-base Coriolis matrix $\mathbf{C}_{\mathrm{a}_k}$ are recalled with the new notations, to be used in the next computations:

$$\begin{aligned} \widehat{\mathbf{M}}_{\mathrm{a}_k} &= {}^{k}\mathbf{N}_{k,l}^\top \left[\mathbf{M}_{\mathrm{a}_k}\right] {}^{k}\mathbf{N}_{k,l} && \text{Composite body inertias of the } k^{th} \text{ appendage} \\ \hat{\dot{\mathbf{M}}}_{\mathrm{a}_k} &= {}^{k}\mathbf{N}_{k,l}^\top \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right] {}^{k}\mathbf{N}_{k,l} && \text{Coriolis and centrifugal matrix of the segments} \\ \mathbf{H}_{\mathrm{a}_k} &= {}^{k}\mathbf{N}_{k,l}^\top \left[\mathbf{M}_{\mathrm{a}_k}\right] {}^{k}\dot{\mathbf{N}}_{k,l} && \text{Intermediate matrix for the computation of } \mathbf{C} \end{aligned}$$

such that the Coriolis matrix reads:

$$\begin{aligned}\mathbf{C}_{\mathrm{a}_k} &= {}^{k}\mathbf{N}_{d}^\top \left({}^{k}\mathbf{N}_{k,l}^\top \left[\mathbf{M}_{\mathrm{a}_k}\right] {}^{k}\dot{\mathbf{N}}_{k,l} + {}^{k}\mathbf{N}_{k,l}^\top \left[\mathbf{M}_{\mathrm{a}_k}\right] {}^{k}\mathbf{N}_{k,l} \left[\boldsymbol{\Omega}_{\mathrm{a}_k}\right] + {}^{k}\mathbf{N}_{k,l}^\top \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right] {}^{k}\mathbf{N}_{k,l}\right) {}^{k}\mathbf{N}_{d} \\ &= {}^{k}\mathbf{N}_{d}^\top \left(\mathbf{H}_{\mathrm{a}_k} + \widehat{\mathbf{M}}_{\mathrm{a}_k} \left[\boldsymbol{\Omega}_{\mathrm{a}_k}\right] + \hat{\dot{\mathbf{M}}}_{\mathrm{a}_k}\right) {}^{k}\mathbf{N}_{d}\end{aligned}$$

Similar to the own mass matrix of the appendages, the diagonal sub-matrices $\mathbf{C}_{\mathrm{a}_k}$ correspond to the computation of a fixed-base appendage, as done in appendix A.3.3. Parts of this algorithm are used to compute the coupling terms ($\mathbf{C}_{\mathrm{ba}_k}$, $\mathbf{C}_{\mathrm{a}_k\mathrm{b}}$) and the spacecraft block $\mathbf{C}_{\mathrm{bb}}$.

The coupling terms are developed using the matrices given in the reminder:

$$\begin{aligned}
\mathbf{C}_{\mathrm{ba}_k} &= {}^k\mathbf{N}_{\mathrm{b}}^{\top}\left(\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^k\dot{\mathbf{N}}_k + \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^k\mathbf{N}_k\right)\\
&= {}^{\mathrm{b}}\mathbf{N}_d^{\top}{}^k\mathbf{N}_{\mathrm{b},l}^{\top}\left(\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^k\dot{\mathbf{N}}_{k,l}{}^k\mathbf{N}_d + \left[\mathbf{M}_{\mathrm{a}_k}\right]{}^k\mathbf{N}_{k,l}{}^k\dot{\mathbf{N}}_d + \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^k\mathbf{N}_{k,l}{}^k\mathbf{N}_d\right)\\
&= {}^{\mathrm{b}}\mathbf{N}_d^{\top}{}^k\mathbf{A}_{\mathrm{b}}^{\top}{}^k\mathbf{N}_{k,l}^{\top}\left(\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^k\dot{\mathbf{N}}_{k,l} + \left[\mathbf{M}_{\mathrm{a}_k}\right]{}^k\mathbf{N}_{k,l}\left[\boldsymbol{\Omega}_{\mathrm{a}_k}\right] + \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^k\mathbf{N}_{k,l}\right){}^k\mathbf{N}_d\\
&= \mathbf{P}_{\mathrm{b}}^{\top}{}^k\mathbf{A}_{\mathrm{b}}^{\top}\left(\mathbf{H}_{\mathrm{a}_k} + \widehat{\mathbf{M}}_{\mathrm{a}_k}\left[\boldsymbol{\Omega}_{\mathrm{a}_k}\right] + \hat{\dot{\mathbf{M}}}_{\mathrm{a}_k}\right){}^k\mathbf{N}_d
\end{aligned}$$

This last result leads to the following expression of the i_k^{th} sub-block of these coupling terms, based on the expressions of $\left(\mathbf{H}_{\mathrm{a}_k}, \widehat{\mathbf{M}}_{\mathrm{a}_k}, \hat{\dot{\mathbf{M}}}_{\mathrm{a}_k}\right)$ in (A.28):

$$\forall i_k = 1_k \ldots n_k, \quad \mathbf{C}_{\mathrm{ba}_k}(i_k) = \mathbf{P}_{\mathrm{b}}^{\top}\mathbf{A}_{1_k,\mathrm{b}}^{\top}\left(\mathbf{A}_{i_k,1_k}^{\top}\mathbf{H}_{i_k} + \mathbf{A}_{i_k,1_k}^{\top}\widehat{\mathbf{M}}_{i_k}\left[\boldsymbol{\Omega}_{i_k}\right] + \mathbf{A}_{i_k,1_k}^{\top}\hat{\dot{\mathbf{M}}}_{i_k}\right)\mathbf{P}_{i_k} \tag{C.9}$$

In the same fashion, the remaining coupling terms read:

$$\begin{aligned}
\mathbf{C}_{\mathrm{a}_k\mathrm{b}} &= {}^k\mathbf{N}_k^{\top}\left(\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^k\dot{\mathbf{N}}_{\mathrm{b}} + \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^k\mathbf{N}_{\mathrm{b}}\right)\\
&= {}^k\mathbf{N}_d^{\top}{}^k\mathbf{N}_{k,l}^{\top}\left(\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^k\dot{\mathbf{N}}_{\mathrm{b},l}{}^{\mathrm{b}}\mathbf{N}_d + \left[\mathbf{M}_{\mathrm{a}_k}\right]{}^k\mathbf{N}_{\mathrm{b},l}{}^{\mathrm{b}}\dot{\mathbf{N}}_d + \left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^k\mathbf{N}_{\mathrm{b},l}{}^{\mathrm{b}}\mathbf{N}_d\right)\\
&= {}^k\mathbf{N}_d^{\top}\left({}^k\mathbf{N}_{k,l}^{\top}\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^k\dot{\mathbf{N}}_{\mathrm{b},l}{}^{\mathrm{b}}\mathbf{N}_d + {}^k\mathbf{N}_{k,l}^{\top}\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^k\mathbf{N}_{\mathrm{b},l}{}^{\mathrm{b}}\dot{\mathbf{N}}_d + {}^k\mathbf{N}_{k,l}^{\top}\left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^k\mathbf{N}_{\mathrm{b},l}{}^{\mathrm{b}}\mathbf{N}_d\right)\\
&= {}^k\mathbf{N}_d^{\top}\Big({}^k\mathbf{N}_{k,l}^{\top}\left[\mathbf{M}_{\mathrm{a}_k}\right]\left({}^k\dot{\mathbf{N}}_{k,l}{}^k\mathbf{A}_{\mathrm{b}} + {}^k\mathbf{N}_{k,l}{}^k\dot{\mathbf{A}}_{\mathrm{b}}\right){}^{\mathrm{b}}\mathbf{N}_d \ \ldots\\
&\qquad\qquad \ldots + {}^k\mathbf{N}_{k,l}^{\top}\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^k\mathbf{N}_{k,l}{}^k\mathbf{A}_{\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}}{}^{\mathrm{b}}\mathbf{N}_d + {}^k\mathbf{N}_{k,l}^{\top}\left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^k\mathbf{N}_{k,l}{}^k\mathbf{A}_{\mathrm{b}}{}^{\mathrm{b}}\mathbf{N}_d\Big)\\
&= {}^k\mathbf{N}_d^{\top}\left(\mathbf{H}_{\mathrm{a}_k}{}^k\mathbf{A}_{\mathrm{b}} + \widehat{\mathbf{M}}_{\mathrm{a}_k}\left({}^k\dot{\mathbf{A}}_{\mathrm{b}} + {}^k\mathbf{A}_{\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}}\right) + \hat{\dot{\mathbf{M}}}_{\mathrm{a}_k}{}^k\mathbf{A}_{\mathrm{b}}\right)\mathbf{P}_{\mathrm{b}}
\end{aligned}$$

leading to:

$$\begin{aligned}
\forall i_k = 1_k \ldots n_k, \quad \mathbf{C}_{\mathrm{a}_k\mathrm{b}}(i_k) =& \mathbf{P}_{i_k}^{\top}\Big(\left(\widehat{\mathbf{M}}_{i_k}\dot{\mathbf{A}}_{i_k,1_k} + \mathbf{H}_{i_k}\mathbf{A}_{i_k,1_k}\right)\mathbf{A}_{1_k,\mathrm{b}}\\
&+ \widehat{\mathbf{M}}_{i_k}\mathbf{A}_{i_k,1_k}\left(\boldsymbol{\Omega}_{\mathrm{b}}\mathbf{A}_{1_k,\mathrm{b}} + \mathbf{A}_{1_k,\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}}\right)\\
&+ \hat{\dot{\mathbf{M}}}_{i_k}\mathbf{A}_{i_k,1_k}\mathbf{A}_{1_k,\mathrm{b}}\Big)\mathbf{P}_{\mathrm{b}}
\end{aligned} \tag{C.10}$$

Regarding the spacecraft term $\mathbf{C}_{\mathrm{bb}}$, its analytic expression is split into two terms. Firstly:

$$\begin{aligned}{}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}}^{\top}\left(\mathbf{M}_{\mathrm{b}}\,{}^{\mathrm{b}}\dot{\mathbf{N}}_{\mathrm{b}}+\dot{\mathbf{M}}_{\mathrm{b}}\,{}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}}\right)&=\mathbf{P}_{\mathrm{b}}^{\top}\left(\mathbf{M}_{\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}}\mathbf{P}_{\mathrm{b}}+\dot{\mathbf{M}}_{\mathrm{b}}\mathbf{P}_{\mathrm{b}}\right)\\&=\mathbf{P}_{\mathrm{b}}^{\top}\left(\mathbf{M}_{\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}}+\dot{\mathbf{M}}_{\mathrm{b}}\right)\mathbf{P}_{\mathrm{b}}\end{aligned}$$

Secondly, an expression similar to the sub-blocks of $\mathbf{C}_{\mathrm{a}_k}$ in (A.28d) and (A.28e) is obtained:

$$\begin{aligned}&\textstyle\sum_k {}^{k}\mathbf{N}_{\mathrm{b}}^{\top}\left(\left[\mathbf{M}_{\mathrm{a}_k}\right]{}^{k}\dot{\mathbf{N}}_{\mathrm{b}}+\left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^{k}\mathbf{N}_{\mathrm{b}}\right)\\=&\textstyle\sum_k {}^{\mathrm{b}}\mathbf{N}_{d}^{\top}{}^{k}\mathbf{N}_{\mathrm{b},l}^{\top}\left(\left[\mathbf{M}_{\mathrm{a}_k}\right]\left({}^{k}\dot{\mathbf{N}}_{\mathrm{b},l}\,{}^{\mathrm{b}}\mathbf{N}_{d}+{}^{k}\mathbf{N}_{\mathrm{b},l}\,{}^{\mathrm{b}}\dot{\mathbf{N}}_{d}\right)+\left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^{k}\mathbf{N}_{\mathrm{b},l}\,{}^{\mathrm{b}}\mathbf{N}_{d}\right)\\=&{}^{\mathrm{b}}\mathbf{N}_{d}^{\top}\left(\textstyle\sum_k {}^{k}\mathbf{N}_{\mathrm{b},l}^{\top}\left(\left[\mathbf{M}_{\mathrm{a}_k}\right]\left({}^{k}\dot{\mathbf{N}}_{\mathrm{b},l}+{}^{k}\mathbf{N}_{\mathrm{b},l}\boldsymbol{\Omega}_{\mathrm{b}}\right)+\left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^{k}\mathbf{N}_{\mathrm{b},l}\right)\right){}^{\mathrm{b}}\mathbf{N}_{d}\\=&\mathbf{P}_{\mathrm{b}}^{\top}\left(\textstyle\sum_k {}^{k}\mathbf{A}_{\mathrm{b}}^{\top}{}^{k}\mathbf{N}_{k,l}^{\top}\left(\left[\mathbf{M}_{\mathrm{a}_k}\right]\left({}^{k}\dot{\mathbf{N}}_{k,l}\,{}^{k}\mathbf{A}_{\mathrm{b}}+{}^{k}\mathbf{N}_{k,l}\,{}^{k}\dot{\mathbf{A}}_{\mathrm{b}}+{}^{k}\mathbf{N}_{k,l}\,{}^{k}\mathbf{A}_{\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}}\right)+\left[\dot{\mathbf{M}}_{\mathrm{a}_k}\right]{}^{k}\mathbf{N}_{k,l}\,{}^{k}\mathbf{A}_{\mathrm{b}}\right)\right)\mathbf{P}_{\mathrm{b}}\\=&\mathbf{P}_{\mathrm{b}}^{\top}\left(\textstyle\sum_k {}^{k}\mathbf{A}_{\mathrm{b}}^{\top}\left(\mathbf{H}_{\mathrm{a}_k}\,{}^{k}\mathbf{A}_{\mathrm{b}}+\widehat{\mathbf{M}}_{\mathrm{a}_k}\left({}^{k}\dot{\mathbf{A}}_{\mathrm{b}}+{}^{k}\mathbf{A}_{\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}}\right)+\hat{\dot{\mathbf{M}}}_{\mathrm{a}_k}\,{}^{k}\mathbf{A}_{\mathrm{b}}\right)\right)\mathbf{P}_{\mathrm{b}}\\=&\mathbf{P}_{\mathrm{b}}^{\top}\left(\textstyle\sum_k \mathbf{A}_{1_k,\mathrm{b}}^{\top}\left(\mathbf{H}_{1_k}\mathbf{A}_{1_k,\mathrm{b}}+\widehat{\mathbf{M}}_{1_k}\left(\dot{\mathbf{A}}_{1_k,\mathrm{b}}+\mathbf{A}_{1_k,\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}}\right)+\hat{\dot{\mathbf{M}}}_{1_k}\mathbf{A}_{1_k,\mathrm{b}}\right)\right)\mathbf{P}_{\mathrm{b}}\end{aligned}$$

By gathering these results, the computation of $\mathbf{C}$ is derived for the whole space robot as follows:

At the appendage level, $\forall k = 1 \dots n_{\mathrm{a}}$,

$$\forall i_k = n_k \dots 1_k,\quad \widehat{\mathbf{M}}_{i_k} = \mathbf{M}_{i_k} + \mathbf{A}_{i_k+1,i_k}^{\top}\,\widehat{\mathbf{M}}_{i_k+1}\,\mathbf{A}_{i_k+1,i_k} \tag{C.11a}$$

$$\hat{\dot{\mathbf{M}}}_{i_k} = \dot{\mathbf{M}}_{i_k} + \mathbf{A}_{i_k+1,i_k}^{\top}\,\hat{\dot{\mathbf{M}}}_{i_k+1}\,\mathbf{A}_{i_k+1,i_k} \tag{C.11b}$$

$$\mathbf{H}_{i_k} = \mathbf{A}_{i_k+1,i_k}^{\top}\left(\mathbf{H}_{i_k+1}\mathbf{A}_{i_k+1,i_k} + \widehat{\mathbf{M}}_{i_k+1}\dot{\mathbf{A}}_{i_k+1,i_k}\right) \tag{C.11c}$$

$$\forall j_k < i_k,\quad \mathbf{C}_{\mathrm{a}_k}(i_k,j_k) = \mathbf{P}_{i_k}^{\top}\left(\left(\mathbf{H}_{i_k} + \hat{\dot{\mathbf{M}}}_{i_k}\right)\mathbf{A}_{i_k,j_k} + \widehat{\mathbf{M}}_{i_k}\left(\dot{\mathbf{A}}_{i_k,j_k} + \mathbf{A}_{i_k,j_k}\boldsymbol{\Omega}_{j_k}\right)\right)\mathbf{P}_{j_k} \tag{C.11d}$$

$$\forall j_k \geq i_k,\quad \mathbf{C}_{\mathrm{a}_k}(i_k,j_k) = \mathbf{P}_{i_k}^{\top}\mathbf{A}_{j_k,i_k}^{\top}\left(\widehat{\mathbf{H}}_{j_k} + \widehat{\mathbf{M}}_{j_k}\boldsymbol{\Omega}_{j_k} + \hat{\dot{\mathbf{M}}}_{j_k}\right)\mathbf{P}_{j_k} \tag{C.11e}$$

$$\mathbf{C}_{\mathrm{ba}_k}(i_k) = \mathbf{P}_{\mathrm{b}}^{\top}\mathbf{A}_{1_k,\mathrm{b}}^{\top}\left(\mathbf{A}_{i_k,1_k}^{\top}\mathbf{H}_{i_k} + \mathbf{A}_{i_k,1_k}^{\top}\widehat{\mathbf{M}}_{i_k}\boldsymbol{\Omega}_{i_k} + \mathbf{A}_{i_k,1_k}^{\top}\hat{\dot{\mathbf{M}}}_{i_k}\right)\mathbf{P}_{i_k} \tag{C.11f}$$

$$\begin{aligned}\mathbf{C}_{\mathrm{a}_k\mathrm{b}}(i_k) = \mathbf{P}_{i_k}^{\top}\Big(&(\widehat{\mathbf{M}}_{i_k}\dot{\mathbf{A}}_{i_k,1_k} + (\mathbf{H}_{i_k} + \hat{\dot{\mathbf{M}}}_{i_k})\mathbf{A}_{i_k,1_k})\mathbf{A}_{1_k,\mathrm{b}}\ \dots\\&\dots\ + \widehat{\mathbf{M}}_{i_k}\mathbf{A}_{i_k,1_k}\left(\boldsymbol{\Omega}_{\mathrm{b}}\mathbf{A}_{1_k,\mathrm{b}} + \mathbf{A}_{1_k,\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}}\right)\Big)\mathbf{P}_{\mathrm{b}}\end{aligned} \tag{C.11g}$$

At the spacecraft level,

$$\begin{aligned}\mathbf{C}_{\mathrm{bb}}(i_k) = &\,\mathbf{P}_{\mathrm{b}}^{\top}\left(\mathbf{M}_{\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}} + \boldsymbol{\Omega}_{\mathrm{b}}\mathbf{M}_{\mathrm{b}}\mathbf{E}_{v}\right)\mathbf{P}_{\mathrm{b}}\dots\\+\,&\mathbf{P}_{\mathrm{b}}^{\top}\left[\textstyle\sum_k \mathbf{A}_{1_k,\mathrm{b}}^{\top}\left(\left(\mathbf{H}_{1_k} + \hat{\dot{\mathbf{M}}}_{1_k}\right)\mathbf{A}_{1_k,\mathrm{b}} + \widehat{\mathbf{M}}_{1_k}\left(\dot{\mathbf{A}}_{1_k,\mathrm{b}} + \mathbf{A}_{1_k,\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}}\right)\right)\right]\mathbf{P}_{\mathrm{b}}\end{aligned} \tag{C.11h}$$

As seen from this set of equations, the algorithm to compute the Coriolis and centrifugal matrix is an extended version of the fixed-base one in Algorithm A.2. The analytic algorithm in Algorithm C.2 provides an insight into the structure of the nonlinear terms, and would be more suited to a symbolic computation. Conversely, the technique based on the Newton-Euler algorithm is more efficient numerically since it is built on vector operations.

Algorithm C.2: Computation of the Coriolis and centrifugal vector for a space robot

Function : $\mathbf{h} = \texttt{CorMatSpace}(\mathbf{q}, \dot{\mathbf{q}})$

Data : $\left\{ {}^{i_k-1}R^{\mathrm{r/f}}_{i_k}\,,\ \mathbf{A}^{\mathrm{r/f}}_{i_k,i_k-1}\,,\ \mathbf{P}_{i_k}\,,\ \mathbf{M}_{i_k}\,,\ \mathbf{A}_{0_k,\mathrm{b}}\,,\ \mathbf{M}_{\mathrm{b}}\,,\ \mathbf{P}_{\mathrm{b}}\ ;\ i_k = 0_k \dots n_k + 1\,,\ k = 1 \dots n_{\mathrm{a}} \right\}$

`Initialize` $\left(\mathbf{H}_{\mathrm{a}} = \mathbf{0}_{6\times 6}\right)$ // Spacecraft Kinetics

$\mathbf{t}_{\mathrm{b}} = \mathbf{P}_{\mathrm{b}}\,\dot{\mathbf{q}}_{\mathrm{b}}$ (6.20)

$\dot{\mathbf{A}}_{0_k,\mathrm{b}} = {}^{0_k}\boldsymbol{R}^{\top}_{\mathrm{b}}\left(\boldsymbol{\Omega}_{\mathrm{b}}\mathbf{A}^{\mathrm{r}}_{0_k,\mathrm{b}} - \mathbf{A}^{\mathrm{r}}_{0_k,\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}}\right)$ (6.17) and (6.18)

for $k = 1 \dots n_{\mathrm{a}}$ **do** // **APPENDAGE TERMS $\mathbf{C}_{\mathrm{a}_K}$ AND $\mathbf{C}_{\mathrm{ba}_K$**

`Initialize` $\left(\mathbf{t}_{0_k} = \mathbf{A}_{0_k,\mathrm{b}}\mathbf{t}_{\mathrm{b}}\right)$ (6.16)

for $i_k = 1_k \dots n_k + 1$ **do** // Recursive computation and storage of $\mathbf{A}_{i_k,i_k-1}$, $\dot{\mathbf{A}}_{i_k,i_k-1}$, and $\boldsymbol{\Omega}_{i_k}$

Kinetics loop from `InvDynSpace` (only $\dot{\mathbf{t}}_{i_k}$ is not computed here)

`Initialize` $\left(\widehat{\mathbf{M}}_{n_k+1} = \mathbf{M}_{n_k+1},\ \hat{\dot{\mathbf{M}}}_{n_k+1} = \boldsymbol{\Omega}_{n_k+1}\mathbf{M}_{n_k+1}\mathbf{E}_v,\ \widehat{\mathbf{H}}_{n_k+1} = \mathbf{0}_{6\times 6}\right)$

for $i_k = n_k \dots 1_k$ **do** // Recursive computation of $\widehat{\mathbf{M}}_{i_k}, \hat{\dot{\mathbf{M}}}_{i_k}, \widehat{\mathbf{H}}_{i_k}$

$\widehat{\mathbf{M}}_{i_k} = \mathbf{M}_{i_k} + \mathbf{A}^{\top}_{i_k+1,i_k}\,\widehat{\mathbf{M}}_{i_k+1}\,\mathbf{A}_{i_k+1,i_k}$ (C.11a)

$\hat{\dot{\mathbf{M}}}_{i_k} = \dot{\mathbf{M}}^{\mathrm{r/f}}_{i_k} + \mathbf{A}^{\top}_{i_k+1,i_k}\,\hat{\dot{\mathbf{M}}}_{i_k+1}\,\mathbf{A}_{i_k+1,i_k}$ (6.38) and (C.11b)

$\mathbf{H}_{i_k} = \mathbf{A}^{\top}_{i_k+1,i_k}\left(\widehat{\mathbf{M}}_{i_k+1}\,\dot{\mathbf{A}}_{i_k+1,i_k} + \mathbf{H}_{i_k+1}\,\mathbf{A}_{i_k+1,i_k}\right)$ (C.11c)

for $j_k = i_k - 1 \dots 1_k$ **do** // Computation of the LHS of line $\mathbf{C}_{\mathrm{a}_k}(i_k, :)$

$\mathbf{A}_{i_k,j_k} = \mathbf{A}_{i_k,j_k+1}\,\mathbf{A}_{j_k+1,j_k}$

$\dot{\mathbf{A}}_{i_k,j_k} = \dot{\mathbf{A}}_{i_k,j_k+1}\,\mathbf{A}_{j_k+1,j_k} + \mathbf{A}_{i_k,j_k+1}\,\dot{\mathbf{A}}_{j_k+1,j_k}$

$\mathbf{C}_{\mathrm{a}_k}(i_k, j_k) = \mathbf{P}^{\top}_{i_k}\left(\left(\widehat{\mathbf{H}}_{i_k} + \hat{\dot{\mathbf{M}}}_{i_k}\right)\mathbf{A}_{i_k,j_k} + \widehat{\mathbf{M}}_{i_k}\left(\dot{\mathbf{A}}_{i_k,j_k} + \mathbf{A}_{i_k,j_k}\boldsymbol{\Omega}_{j_k}\right)\right)\mathbf{P}_{j_k}$ (C.11d)

for $j_k = i_k \dots n_k$ **do** // Computation of RHS of line $\mathbf{C}_{\mathrm{a}_k}(i_k, :)$

$\mathbf{C}_{\mathrm{a}_k}(i_k, j_k) = \mathbf{P}^{\top}_{i_k}\mathbf{A}^{\top}_{j_k,i_k}\left(\widehat{\mathbf{H}}_{j_k} + \widehat{\mathbf{M}}_{j_k}\boldsymbol{\Omega}_{j_k} + \hat{\dot{\mathbf{M}}}_{j_k}\right)\mathbf{P}_{j_k}$ (C.11e)

$\mathbf{A}_{j_k+1,i_k} = \mathbf{A}_{j_k+1,j_k}\,\mathbf{A}_{j_k,i_k}$

$\mathbf{A}_{1_k,\mathrm{b}} = \mathbf{A}_{1_k,0_k}, \mathbf{A}_{0_k,\mathrm{b}}$ // Computation of $\mathbf{C}_{\mathrm{ba}_k}(i_k)$

$\mathbf{C}_{\mathrm{ba}_k}(i_k) = \dots$ and $\mathbf{C}_{\mathrm{a}_k\mathrm{b}}(i_k) = \dots$ (C.11f) and (C.11g)

$\dot{\mathbf{A}}_{1_k,\mathrm{b}} = \dot{\mathbf{A}}_{1_k,0_k}, \mathbf{A}_{0_k,\mathrm{b}} + \mathbf{A}_{1_k,0_k}, \dot{\mathbf{A}}_{0_k,\mathrm{b}}$ // Intermediate variable to compute $\mathbf{C}_{\mathrm{bb}}$

$\mathbf{H}_{\mathrm{a}} = \mathbf{H}_{\mathrm{a}} + \mathbf{A}^{\top}_{1_k,\mathrm{b}}\left(\mathbf{H}_{1_k}\mathbf{A}_{1_k,\mathrm{b}} + \widehat{\mathbf{M}}_{1_k}\left(\dot{\mathbf{A}}_{1_k,\mathrm{b}} + \mathbf{A}_{1_k,\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}}\right) + \hat{\dot{\mathbf{M}}}_{1_k}\mathbf{A}_{1_k,\mathrm{b}}\right)$ (C.11h)

// **SPACECRAFT TERM $\mathbf{C}_{\mathrm{bb}}$**

$\mathbf{C}_{\mathrm{bb}} = \mathbf{P}^{\top}_{\mathrm{b}}\left(\mathbf{M}_{\mathrm{b}}\boldsymbol{\Omega}_{\mathrm{b}} + \dot{\mathbf{M}}_{\mathrm{b}} + \mathbf{H}\mathrm{a}\right)\mathbf{P}_{\mathrm{b}}$ (C.11h)

return $\mathbf{h} = \mathbf{C}\,\dot{\mathbf{q}}$

C.1.3 Forward Dynamics Algorithm

The forward dynamics algorithm is now derived for a space robot with a star shape. The steps presented in Algorithm A.3 for one appendage on a fixed base are extended to a multi-appendages approach. As introduced in the main content, the DeNOC matrix $\mathbf{N}$ for tree-type systems can be decoupled in a form similar to the chain-like manipulators one, with identity matrix on the diagonal. This decomposition considers each appendage as a "module", that replaces the previous concept of segment. As assumed with the star shape, each of these modules is itself a chain-like system. With this bottom-up approach, the same algorithm of inversion is applied on the global mass matrix and results in going back and forth between the multi-module algorithm and the previous mono-module one. These switches between both visions are summarized in the general case of tree-type system in (Shah et al., 2012a).

Reminder: It is shown in (6.57) how the DeNOC matrix $\mathbf{N}$ can be written in a form similar to the manipulator case with:

$$\mathbf{N} = \underbrace{\left[\begin{array}{c|cccc} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline {}^{1}\mathbf{N}_{\mathrm{b},l} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ {}^{2}\mathbf{N}_{\mathrm{b},l} & \mathbf{0} & \mathbf{I} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ {}^{n_\mathrm{a}}\mathbf{N}_{\mathrm{b},l} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{I} \end{array}\right]}_{=\mathbf{N}_l} \underbrace{\left[\begin{array}{ccccc} {}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^{1}\mathbf{N}_{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & {}^{2}\mathbf{N}_{2} & \ddots & \vdots \\ \mathbf{0} & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & {}^{n_\mathrm{a}}\mathbf{N}_{n_\mathrm{a}} \end{array}\right]}_{=\mathbf{N}_d} \tag{C.12}$$

With this decomposition, a parallel can be made with the DeNOC matrices $\mathbf{N}_l$ and $\mathbf{N}_d$ of a manipulator in (4.32), by defining the following equivalences, where the index 0 is replaced by the spacecraft notation b:

$$\begin{aligned} \mathbf{P}_k &\Longrightarrow {}^{k}\mathbf{N}_{k} \\ \mathbf{A}_{k,j} &\Longrightarrow \mathbf{0} \quad \forall k = 1 \dots n_\mathrm{a},\ \forall j = 1 \dots k-1 \\ \mathbf{A}_{k,\mathrm{b}} &\Longrightarrow {}^{k}\mathbf{N}_{\mathrm{b},l} \\ \mathbf{M}_k &\Longrightarrow [\mathbf{M}_{\mathrm{a}_k}] \end{aligned}$$

Reminding the $\mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top$ decomposition of the mass matrix proposed in Appendix A.3.4, the upper triangular matrix $\mathbf{U}$ reads as follows for our star-shaped system:

$$\mathbf{U} = \left[\begin{array}{c|cccc} \mathbf{I} & \mathbf{U}_{\mathrm{b},1} & \cdots & \cdots & \mathbf{U}_{\mathrm{b},n_\mathrm{a}} \\ \hline \mathbf{0} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{I} \end{array}\right] \quad \boldsymbol{\Delta} = \left[\begin{array}{c|ccc} \boldsymbol{\Delta}_\mathrm{b} & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \mathbf{0} & \boldsymbol{\Delta}_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\Delta}_{n_\mathrm{a}} \end{array}\right]$$

where the sparse definition of $\mathbf{N}_l$ implies the same structure for $\mathbf{U}$. By definition of $\mathbf{U}$ and $\boldsymbol{\Delta}$, the global mass matrix decomposes at the multi-module level by:

$$\mathbf{D} = \mathbf{U}\,\boldsymbol{\Delta}\,\mathbf{U}^\top$$

With the module equivalence, the explicit expressions of matrices $\mathbf{U}_{i,j} = \mathbf{P}_i^\top \mathbf{A}_{j,i}^\top \boldsymbol{\varphi}_j$ and $\boldsymbol{\Delta}_i = \mathbf{P}_i^\top \bar{\mathbf{M}}_i \mathbf{P}_i$ in (A.33) and (A.37), are updated by:

$$\bar{\mathbf{M}}_k = [\mathbf{M}_{\mathrm{a}_k}] \tag{C.13}$$

$$\boldsymbol{\Delta}_k = {}^k\mathbf{N}_k^\top \bar{\mathbf{M}}_k {}^k\mathbf{N}_k = {}^k\mathbf{N}_d^\top \widehat{\mathbf{M}}_{\mathrm{a}_k} {}^k\mathbf{N}_d \tag{C.14}$$

$$\mathbf{U}_{j,k} = \mathbf{0} \quad \forall k = 1 \dots n_\mathrm{a},\, \forall j = 1 \dots k-1 \tag{C.15}$$

$$\mathbf{U}_{\mathrm{b},k} = {}^\mathrm{b}\mathbf{N}_\mathrm{b}^\top {}^k\mathbf{N}_{\mathrm{b},l}^\top \bar{\mathbf{M}}_k {}^k\mathbf{N}_k \boldsymbol{\Delta}_k^{-1} \tag{C.16}$$

Then the three steps of forward dynamics are applied with these module matrices. The whole system $\left(\mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top\right)\mathbf{X} = \mathbf{b}$ is solved in the same fashion by:

1. Solve: $\mathbf{U}\widehat{\mathbf{X}} = \mathbf{b}$
2. Solve: $\boldsymbol{\Delta}\bar{\mathbf{X}} = \widehat{\mathbf{X}}$ $(= \mathbf{U}^{-1}\,\mathbf{b})$
3. Solve: $\mathbf{U}^\top\mathbf{X} = \bar{\mathbf{X}}$ $(= \boldsymbol{\Delta}^{-1}\,\mathbf{U}^{-1}\,\mathbf{b})$

Step 1: The system $\mathbf{U}\widehat{\mathbf{X}} = \mathbf{b}$ is easily solved due to the sparse structure of $\mathbf{U}$. It reads as follows:

$$\left[\begin{array}{c|cccc} \mathbf{I} & \mathbf{U}_{\mathrm{b},1} & \cdots & \cdots & \mathbf{U}_{\mathrm{b},n_\mathrm{a}} \\ \hline \mathbf{0} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \mathbf{0} & \mathbf{I} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{array}\right] \left[\begin{array}{c} \widehat{\mathbf{X}}_\mathrm{b} \\ \hline \widehat{\mathbf{X}}_1 \\ \vdots \\ \widehat{\mathbf{X}}_{n_\mathrm{a}} \end{array}\right] = \left[\begin{array}{c} \mathbf{b}_\mathrm{b} \\ \hline \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_{n_\mathrm{a}} \end{array}\right]$$

and its solution is direct for the lower terms with $\widehat{\mathbf{X}}_k = \mathbf{b}_k$, but not for the spacecraft terms $\widehat{\mathbf{X}}_\mathrm{b}$:

$$\begin{aligned}
\widehat{\mathbf{X}}_\mathrm{b} &= \mathbf{b}_\mathrm{b} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{U}_{\mathrm{b},k}\, \widehat{\mathbf{X}}_k \\
&= \mathbf{b}_\mathrm{b} - \sum_{k=1}^{n_\mathrm{a}} {}^{\mathrm{b}}\mathbf{N}_\mathrm{b}^\top\, {}^{k}\mathbf{N}_{\mathrm{b},l}^\top\, \bar{\mathbf{M}}_k\, {}^{k}\mathbf{N}_k\, \boldsymbol{\Delta}_k^{-1}\, \mathbf{b}_k \\
&= \mathbf{b}_\mathrm{b} - \sum_{k=1}^{n_\mathrm{a}} {}^{\mathrm{b}}\mathbf{N}_d^\top\, {}^{k}\mathbf{A}_\mathrm{b}^\top\, {}^{k}\mathbf{N}_{k,l}\, [\mathbf{M}_{\mathrm{a}_k}]\, {}^{k}\mathbf{N}_{k,l}\, {}^{k}\mathbf{N}_d\, \boldsymbol{\Delta}_k^{-1}\, \mathbf{b}_k \\
&= \mathbf{b}_\mathrm{b} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{P}_\mathrm{b}^\top\, {}^{k}\mathbf{A}_\mathrm{b}^\top\, \widehat{\mathbf{M}}_{\mathrm{a}_k}\, {}^{k}\mathbf{N}_d\, \boldsymbol{\Delta}_k^{-1}\, \mathbf{b}_k
\end{aligned}$$

Using now the $\mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top$ decomposition of the module itself, using Appendix A.3.4, $\boldsymbol{\Delta}_k$ and $\widehat{\mathbf{M}}_{\mathrm{a}_k}$ are expanded as follows:

$$\begin{aligned}
\boldsymbol{\Delta}_k &= \mathbf{U}_{\mathrm{a}_k} \boldsymbol{\Delta}_{\mathrm{a}_k} \mathbf{U}_{\mathrm{a}_k}^\top \\
\widehat{\mathbf{M}}_{\mathrm{a}_k} &= \tilde{\mathbf{U}}_{\mathrm{a}_k} \boldsymbol{\Delta}_{\mathrm{a}_k} \tilde{\mathbf{U}}_{\mathrm{a}_k}^\top
\end{aligned}$$

with the definition of $\tilde{\mathbf{U}}_{\mathrm{a}_k}$ in (A.40) by:

$$\mathbf{U}_{\mathrm{a}_k} = {}^{k}\mathbf{N}_d^\top\, \tilde{\mathbf{U}}_{\mathrm{a}_k}$$

The global triangular system is then solved at the spacecraft level by:

$$\begin{aligned}
\widehat{\mathbf{X}}_\mathrm{b} &= \mathbf{b}_\mathrm{b} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{P}_\mathrm{b}^{\top k}\mathbf{A}_\mathrm{b}^\top \widehat{\mathbf{M}}_{\mathrm{a}_k}\, {}^{k}\mathbf{N}_d \boldsymbol{\Delta}_k^{-1} \mathbf{b}_k \\
&= \mathbf{b}_\mathrm{b} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{P}_\mathrm{b}^{\top k}\mathbf{A}_\mathrm{b}^\top \tilde{\mathbf{U}}_{\mathrm{a}_k} \boldsymbol{\Delta}_{\mathrm{a}_k} \tilde{\mathbf{U}}_{\mathrm{a}_k}^\top\, {}^{k}\mathbf{N}_d \mathbf{U}_{\mathrm{a}_k}^{-\top} \boldsymbol{\Delta}_{\mathrm{a}_k}^{-1} \mathbf{U}_{\mathrm{a}_k}^{-1} \mathbf{b}_k \\
&= \mathbf{b}_\mathrm{b} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{P}_\mathrm{b}^{\top k}\mathbf{A}_\mathrm{b}^\top \tilde{\mathbf{U}}_{\mathrm{a}_k} \boldsymbol{\Delta}_{\mathrm{a}_k} \mathbf{U}_{\mathrm{a}_k}^\top \mathbf{U}_{\mathrm{a}_k}^{-\top} \boldsymbol{\Delta}_{\mathrm{a}_k}^{-1} \mathbf{U}_{\mathrm{a}_k}^{-1} \mathbf{b}_k \\
&= \mathbf{b}_\mathrm{b} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{P}_\mathrm{b}^{\top k}\mathbf{A}_\mathrm{b}^\top \tilde{\mathbf{U}}_{\mathrm{a}_k} \mathbf{U}_{\mathrm{a}_k}^{-1} \mathbf{b}_k
\end{aligned}$$

Reminding from Appendix A.3.4 that the intermediate variable obtained after the first step was denoted by: $\widehat{\mathbf{X}} = \mathbf{U}^{-1}\mathbf{b}$, the previous summation can be re-written as:

$$\widehat{\mathbf{X}}_\mathrm{b} = \mathbf{b}_\mathrm{b} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{P}_\mathrm{b}^{\top k}\mathbf{A}_\mathrm{b}^\top \tilde{\mathbf{U}}_{\mathrm{a}_k} \widehat{\mathbf{X}}_{\mathrm{a}_k}$$

where $\widehat{\mathbf{X}}_{\mathrm{a}_k} = \mathbf{U}_{\mathrm{a}_k}^{-1}\mathbf{b}_k$ is solved with the previous algorithm for one appendage on a fixed base. It is worth noticing that this vector is different from $\widehat{\mathbf{X}}_k = \mathbf{b}_k$ obtained from the multi-module solution.

Based on the expression of $\tilde{\mathbf{U}}$ in (A.39), the matrix multiplying $\widehat{\mathbf{X}}_{\mathrm{a}_k}$ can be developed as:

$$\begin{aligned}
{}^{k}\mathbf{A}_{\mathrm{b}}^{\top}\tilde{\mathbf{U}}_{\mathrm{a}_k}\,\widehat{\mathbf{X}}_{\mathrm{a}_k} &= \mathbf{A}_{1_k,\mathrm{b}}^{\top}\begin{bmatrix}\tilde{\mathbf{U}}_{1_k,1_k} & \dots & \tilde{\mathbf{U}}_{1_k,n_k}\end{bmatrix}\widehat{\mathbf{X}}_{\mathrm{a}_k}\\
&= \mathbf{A}_{1_k,\mathrm{b}}^{\top}\begin{bmatrix}\boldsymbol{\varphi}_{1_k} & \dots & \mathbf{A}_{n_k,1_k}^{\top}\,\boldsymbol{\varphi}_{n_k}\end{bmatrix}\widehat{\mathbf{X}}_{\mathrm{a}_k}\\
&= \mathbf{A}_{1_k,\mathrm{b}}^{\top}\underbrace{\sum_{i_k=1_k}^{n_k}\mathbf{A}_{i_k,1_k}^{\top}\,\boldsymbol{\varphi}_{i_k}\widehat{\mathbf{X}}_{i_k}}_{=\boldsymbol{\eta}_{1_k}}
\end{aligned}$$

This main result allows to fully solve the global triangular system by:

$$\forall k = 1\dots n_{\mathrm{a}},\quad \widehat{\mathbf{X}}_k = \mathbf{b}_k \tag{C.17a}$$

$$\widehat{\mathbf{X}}_{\mathrm{b}} = \mathbf{b}_{\mathrm{b}} - \sum_{k=1}^{n_{\mathrm{a}}}\mathbf{P}_{\mathrm{b}}^{\top}\,\mathbf{A}_{1_k,\mathrm{b}}^{\top}\,\boldsymbol{\eta}_{1_k} \tag{C.17b}$$

This scheme is the basis of the forward dynamics for multi-module systems. It allows to solve the first step for $\widehat{\mathbf{X}}$ in the global approach, by computing this same first step at the module level in order to obtain the variables $\boldsymbol{\eta}_{1_k}$. They are then added to get the spacecraft variable $\widehat{\mathbf{X}}_{\mathrm{b}}$ at the multi-module level.

Step 2: The system $\boldsymbol{\Delta}\bar{\mathbf{X}} = \widehat{\mathbf{X}}$ is solved for $\bar{\mathbf{X}}$ provided that **Step 1** yields $\widehat{\mathbf{X}}$:

$$\left[\begin{array}{c|cccc}
\boldsymbol{\Delta}_{\mathrm{b}} & \mathbf{0} & \cdots & \cdots & \mathbf{0}\\ \hline
\mathbf{0} & \boldsymbol{\Delta}_1 & \mathbf{0} & \cdots & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \ddots & \ddots & \vdots\\
\vdots & \vdots & \ddots & \ddots & \mathbf{0}\\
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\Delta}_{n_{\mathrm{a}}}
\end{array}\right]\left[\begin{array}{c}
\bar{\mathbf{X}}_{\mathrm{b}}\\ \hline
\bar{\mathbf{X}}_1\\ \vdots\\ \bar{\mathbf{X}}_{n_{\mathrm{a}}}
\end{array}\right] = \left[\begin{array}{c}
\widehat{\mathbf{X}}_{\mathrm{b}}\\ \hline
\widehat{\mathbf{X}}_1\\ \vdots\\ \widehat{\mathbf{X}}_{n_{\mathrm{a}}}
\end{array}\right]$$

The solution is straightforward, and is given by:

$$\bar{\mathbf{X}}_{\mathrm{b}} = \boldsymbol{\Delta}_{\mathrm{b}}^{-1}\,\widehat{\mathbf{X}}_{\mathrm{b}} \tag{C.18a}$$

$$\forall k = 1\dots n_{\mathrm{a}},\quad \bar{\mathbf{X}}_k = \boldsymbol{\Delta}_k^{-1}\,\widehat{\mathbf{X}}_k \tag{C.18b}$$

It is very important to notice that the appendage solutions are actually obtained by inverting

their whole mass matrix $\boldsymbol{\Delta}_k^{-1}$, which would be very costly to perform. Instead, the algorithm avoids this step thanks to the third step solution, where the values of $\bar{\mathbf{X}}_k$ are not required.

At the spacecraft level though, the matrix $\boldsymbol{\Delta}_\mathrm{b}$ must be computed explicitly. It is defined by the equivalent definition of the mass matrix $\mathbf{D}_\mathrm{bb}$ either by (C.6f), or by the direct product $\mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top$. Its expression satisfies the relation in (A.30), re-written with the module matrices as:

$$\mathbf{D}_\mathrm{bb} = \begin{cases} \boldsymbol{\Delta}_\mathrm{b} + \sum_{k=1}^{n_\mathrm{a}} \mathbf{U}_{\mathrm{b},k} \boldsymbol{\Delta}_k \mathbf{U}_{\mathrm{b},k}^\top \\ \mathbf{P}_\mathrm{b}^\top \left(\mathbf{M}_\mathrm{b} + \sum_k \mathbf{A}_{1_k,\mathrm{b}}^\top \widehat{\mathbf{M}}_{1_k} \mathbf{A}_{1_k,\mathrm{b}} \right) \mathbf{P}_\mathrm{b} \end{cases}$$

This relation leads to:

$$\begin{aligned}
\boldsymbol{\Delta}_\mathrm{b} &= \mathbf{D}_\mathrm{bb} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{U}_{\mathrm{b},k} \boldsymbol{\Delta}_k \mathbf{U}_{\mathrm{b},k}^\top \\
&= \mathbf{D}_\mathrm{bb} - \sum_{k=1}^{n_\mathrm{a}} {}^\mathrm{b}\mathbf{N}_\mathrm{b}^{\top\,k} \mathbf{N}_{\mathrm{b},l}^\top \bar{\mathbf{M}}_k {}^k\mathbf{N}_k \boldsymbol{\Delta}_k^{-1} \boldsymbol{\Delta}_k \boldsymbol{\Delta}_k^{-1\,k} \mathbf{N}_k^\top \bar{\mathbf{M}}_k {}^k\mathbf{N}_{\mathrm{b},l} {}^\mathrm{b}\mathbf{N}_\mathrm{b} \\
&= \mathbf{D}_\mathrm{bb} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{P}_\mathrm{b}^{\top\,k} \mathbf{A}_\mathrm{b}^{\top\,k} \mathbf{N}_{k,l}^\top \left[\mathbf{M}_{\mathrm{a}_k}\right] {}^k\mathbf{N}_{k,l} {}^k\mathbf{N}_d \boldsymbol{\Delta}_k^{-1\,k} \mathbf{N}_d^{\top\,k} \mathbf{N}_{k,l}^\top \left[\mathbf{M}_{\mathrm{a}_k}\right] {}^k\mathbf{N}_{k,l} {}^k\mathbf{A}_\mathrm{b} \mathbf{P}_\mathrm{b} \\
&= \mathbf{D}_\mathrm{bb} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{P}_\mathrm{b}^{\top\,k} \mathbf{A}_\mathrm{b}^{\top\,k} \widehat{\mathbf{M}}_{\mathrm{a}_k} {}^k\mathbf{N}_d \boldsymbol{\Delta}_k^{-1\,k} \mathbf{N}_d^\top \widehat{\mathbf{M}}_{\mathrm{a}_k} \mathbf{A}_\mathrm{b} \mathbf{P}_\mathrm{b}
\end{aligned}$$

using again the own $\mathbf{U}\boldsymbol{\Delta}\mathbf{U}^\top$ decomposition of the appendage inertia, it yields:

$$\begin{aligned}
\boldsymbol{\Delta}_\mathrm{b} &= \mathbf{D}_\mathrm{bb} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{P}_\mathrm{b}^{\top\,k} \mathbf{A}_\mathrm{b}^\top \tilde{\mathbf{U}}_{\mathrm{a}_k} \boldsymbol{\Delta}_{\mathrm{a}_k} \tilde{\mathbf{U}}_{\mathrm{a}_k}^{\top\,k} \mathbf{N}_d \mathbf{U}_{\mathrm{a}_k}^{-\top} \boldsymbol{\Delta}_{\mathrm{a}_k}^{-1} \mathbf{U}_{\mathrm{a}_k}^{-1\,k} \mathbf{N}_d^\top \tilde{\mathbf{U}}_{\mathrm{a}_k} \boldsymbol{\Delta}_{\mathrm{a}_k} \tilde{\mathbf{U}}_{\mathrm{a}_k}^{\top\,k} \mathbf{A}_\mathrm{b} \mathbf{P}_\mathrm{b} \\
&= \mathbf{D}_\mathrm{bb} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{P}_\mathrm{b}^{\top\,k} \mathbf{A}_\mathrm{b}^\top \tilde{\mathbf{U}}_{\mathrm{a}_k} \boldsymbol{\Delta}_{\mathrm{a}_k} \mathbf{U}_{\mathrm{a}_k}^\top \mathbf{U}_{\mathrm{a}_k}^{-\top} \boldsymbol{\Delta}_{\mathrm{a}_k}^{-1} \mathbf{U}_{\mathrm{a}_k}^{-1} \mathbf{U}_{\mathrm{a}_k} \boldsymbol{\Delta}_{\mathrm{a}_k} \tilde{\mathbf{U}}_{\mathrm{a}_k}^{\top\,k} \mathbf{A}_\mathrm{b} \mathbf{P}_\mathrm{b} \\
&= \mathbf{D}_\mathrm{bb} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{P}_\mathrm{b}^{\top\,k} \mathbf{A}_\mathrm{b}^\top \tilde{\mathbf{U}}_{\mathrm{a}_k} \boldsymbol{\Delta}_{\mathrm{a}_k} \tilde{\mathbf{U}}_{\mathrm{a}_k}^{\top\,k} \mathbf{A}_\mathrm{b} \mathbf{P}_\mathrm{b} \\
&= \mathbf{D}_\mathrm{bb} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{P}_\mathrm{b}^\top \mathbf{A}_{1_k,\mathrm{b}}^\top \begin{bmatrix} \boldsymbol{\varphi}_{1_k} & \dots & \mathbf{A}_{n_k,1_k}^\top \boldsymbol{\varphi}_{n_k} \end{bmatrix} \boldsymbol{\Delta}_{\mathrm{a}_k} \begin{bmatrix} \boldsymbol{\varphi}_{1_k} & \dots & \mathbf{A}_{n_k,1_k}^\top \boldsymbol{\varphi}_{n_k} \end{bmatrix}^\top \mathbf{A}_{1_k,\mathrm{b}} \mathbf{P}_\mathrm{b} \\
&= \mathbf{D}_\mathrm{bb} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{P}_\mathrm{b}^\top \mathbf{A}_{1_k,\mathrm{b}}^\top \sum_{i_k=1_k}^{n_k} \mathbf{A}_{i_k,1_k}^\top \boldsymbol{\varphi}_{i_k} \boldsymbol{\Delta}_{i_k} \boldsymbol{\varphi}_{i_k}^\top \mathbf{A}_{i_k,1_k} \mathbf{A}_{1_k,\mathrm{b}} \mathbf{P}_\mathrm{b} \\
&= \mathbf{D}_\mathrm{bb} - \sum_{k=1}^{n_\mathrm{a}} \mathbf{P}_\mathrm{b}^\top \mathbf{A}_{1_k,\mathrm{b}}^\top \underbrace{\sum_{i_k=1_k}^{n_k} \mathbf{A}_{i_k,1_k}^\top \boldsymbol{\varphi}_{i_k} \bar{\boldsymbol{\varphi}}_{i_k}^\top \mathbf{A}_{i_k,1_k}}_{=\widehat{\mathbf{M}}_{1_k} - \bar{\mathbf{M}}_{1_k} + \boldsymbol{\varphi}_{1_k} \bar{\boldsymbol{\varphi}}_{1_k}^\top} \mathbf{A}_{1_k,\mathrm{b}} \mathbf{P}_\mathrm{b}
\end{aligned}$$

using the definition of $\bar{\mathbf{M}}_{1_k}$ introduced in the proof of the forward dynamics algorithm in Appendix A.3.4.

Reminding then the expression of $\mathbf{D}_{\mathrm{bb}}$, the result reduces to:

$$\begin{aligned}\boldsymbol{\Delta}_{\mathrm{b}} &= \mathbf{D}_{\mathrm{bb}} + \sum_{k=1}^{n_{\mathrm{a}}} \mathbf{P}_{\mathrm{b}}^{\top} \mathbf{A}_{1_k,\mathrm{b}}^{\top} \left(\bar{\mathbf{M}}_{1_k} - \widehat{\mathbf{M}}_{1_k} - \boldsymbol{\varphi}_{1_k} \bar{\boldsymbol{\varphi}}_{1_k}^{\top}\right) \mathbf{A}_{1_k,\mathrm{b}} \mathbf{P}_{\mathrm{b}} \\ &= \mathbf{P}_{\mathrm{b}}^{\top} \left(\mathbf{M}_{\mathrm{b}} + \sum_{k=1}^{n_{\mathrm{a}}} \mathbf{A}_{1_k,\mathrm{b}}^{\top} \left(\bar{\mathbf{M}}_{1_k} - \boldsymbol{\varphi}_{1_k} \bar{\boldsymbol{\varphi}}_{1_k}^{\top}\right) \mathbf{A}_{1_k,\mathrm{b}}\right) \mathbf{P}_{\mathrm{b}} \end{aligned} \tag{C.19}$$

This expression allows to recursively compute the summation inside $\boldsymbol{\Delta}_{\mathrm{b}}$, using the same first step of the mono-module recursions. Indeed, these latter provides the necessary values of $\left(\bar{\mathbf{M}}_{1_k}, \boldsymbol{\varphi}_{1_k}, \bar{\boldsymbol{\varphi}}_{1_k}\right)$. Once this matrix is obtained, its inversion yields the value of $\bar{\mathbf{X}}_{\mathrm{b}}$ to initialize the third step.

Step 3: Eventually, the system $\mathbf{U}^{\top}\mathbf{X} = \bar{\mathbf{X}}$ is solved using the same modified RGE algorithm on:

$$\left[\begin{array}{c|cccc} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \mathbf{U}_{\mathrm{b},1}^{\top} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \mathbf{0} & \mathbf{I} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{U}_{\mathrm{b},n_{\mathrm{a}}} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{array}\right] \left[\begin{array}{c} \mathbf{X}_{\mathrm{b}} \\ \hline \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_n \end{array}\right] = \left[\begin{array}{c} \bar{\mathbf{X}}_{\mathrm{b}} \\ \hline \bar{\mathbf{X}}_1 \\ \vdots \\ \bar{\mathbf{X}}_{n_{\mathrm{a}}} \end{array}\right]$$

The solution at the spacecraft level is straightforward and yields:

$$\mathbf{X}_{\mathrm{b}} = \bar{\mathbf{X}}_{\mathrm{b}}$$

while the appendages terms are given by:

$$\begin{aligned}\mathbf{X}_k &= \bar{\mathbf{X}}_k - \mathbf{U}_{\mathrm{b},k}^{\top} \mathbf{X}_{\mathrm{b}} \\ &= \bar{\mathbf{X}}_k - \boldsymbol{\Delta}_k^{-1}\, {}^{k}\mathbf{N}_k^{\top}\, \bar{\mathbf{M}}_k\, {}^{k}\mathbf{N}_{\mathrm{b},l}\, {}^{\mathrm{b}}\mathbf{N}_{\mathrm{b}}\, \mathbf{X}_{\mathrm{b}} \\ &= \bar{\mathbf{X}}_k - \boldsymbol{\Delta}_k^{-1}\, {}^{k}\mathbf{N}_d^{\top}\, {}^{k}\mathbf{N}_{k,l}^{\top}\, [\mathbf{M}_{\mathrm{a}_k}]\, {}^{k}\mathbf{N}_{k,l}\, {}^{k}\mathbf{A}_{\mathrm{b}}\, \mathbf{P}_{\mathrm{b}}\, \mathbf{X}_{\mathrm{b}} \\ &= \bar{\mathbf{X}}_k - \boldsymbol{\Delta}_k^{-1}\, {}^{k}\mathbf{N}_d^{\top}\, \widehat{\mathbf{M}}_{\mathrm{a}_k}\, {}^{k}\mathbf{A}_{\mathrm{b}}\, \mathbf{P}_{\mathrm{b}}\, \mathbf{X}_{\mathrm{b}}\end{aligned}$$

Using again the own appendage decomposition, one obtains:

$$\begin{aligned}\mathbf{X}_k &= \bar{\mathbf{X}}_k - \mathbf{U}_{\mathrm{a}_k}^{-\top}\boldsymbol{\Delta}_{\mathrm{a}_k}^{-1}\mathbf{U}_{\mathrm{a}_k}^{-1\,k}\mathbf{N}_d^{\top}\,\tilde{\mathbf{U}}_{\mathrm{a}_k}\boldsymbol{\Delta}_{\mathrm{a}_k}\tilde{\mathbf{U}}_{\mathrm{a}_k}^{\top\,k}\mathbf{A}_{\mathrm{b}}\,\mathbf{P}_{\mathrm{b}}\,\mathbf{X}_{\mathrm{b}} \\ &= \bar{\mathbf{X}}_k - \mathbf{U}_{\mathrm{a}_k}^{-\top}\boldsymbol{\Delta}_{\mathrm{a}_k}^{-1}\mathbf{U}_{\mathrm{a}_k}^{-1}\,\mathbf{U}_{\mathrm{a}_k}\boldsymbol{\Delta}_{\mathrm{a}_k}\tilde{\mathbf{U}}_{\mathrm{a}_k}^{\top\,k}\mathbf{A}_{\mathrm{b}}\,\mathbf{P}_{\mathrm{b}}\,\mathbf{X}_{\mathrm{b}} \\ &= \bar{\mathbf{X}}_k - \mathbf{U}_{\mathrm{a}_k}^{-\top}\tilde{\mathbf{U}}_{\mathrm{a}_k}^{\top\,k}\mathbf{A}_{\mathrm{b}}\,\mathbf{P}_{\mathrm{b}}\,\mathbf{X}_{\mathrm{b}} \\ &= \bar{\mathbf{X}}_k - \mathbf{U}_{\mathrm{a}_k}^{-\top}\tilde{\mathbf{U}}_{\mathrm{a}_k}^{\top\,k}\mathbf{A}_{\mathrm{b}}\,\mathbf{P}_{\mathrm{b}}\,\mathbf{X}_{\mathrm{b}}\end{aligned}$$

Introducing the results of the second step, a change of variable is made with:

$$\begin{aligned}\bar{\mathbf{X}}_k &= \boldsymbol{\Delta}_k^{-1}\,\widehat{\mathbf{X}}_k \\ &= \mathbf{U}_{\mathrm{a}_k}^{-\top}\boldsymbol{\Delta}_{\mathrm{a}_k}^{-1}\mathbf{U}_{\mathrm{a}_k}^{-1}\,\mathbf{b}_k \\ &= \mathbf{U}_{\mathrm{a}_k}^{-\top}\bar{\mathbf{X}}_{\mathrm{a}_k}\end{aligned}$$

such that the global variable $\mathbf{X}_k$ can be written as:

$$\begin{aligned}\mathbf{X}_k &= \mathbf{U}_{\mathrm{a}_k}^{-\top}\left(\bar{\mathbf{X}}_{\mathrm{a}_k} - \tilde{\mathbf{U}}_{\mathrm{a}_k}^{\top\,k}\mathbf{A}_{\mathrm{b}}\,\mathbf{P}_{\mathrm{b}}\,\mathbf{X}_{\mathrm{b}}\right) \\ &= \mathbf{U}_{\mathrm{a}_k}^{-\top}\left(\bar{\mathbf{X}}_{\mathrm{a}_k} - \left(\mathbf{A}_{1_k,\mathrm{b}}^{\top}\left[\begin{array}{ccc}\boldsymbol{\varphi}_{1_k} & \dots & \mathbf{A}_{n_k,1_k}^{\top}\,\boldsymbol{\varphi}_{n_k}\end{array}\right]\right)^{\top}\mathbf{P}_{\mathrm{b}}\,\mathbf{X}_{\mathrm{b}}\right)\end{aligned}$$

The system to solve at the module level is eventually given by:

$$\mathbf{U}_{\mathrm{a}_k}^{\top}\,\mathbf{X}_k = \bar{\mathbf{X}}_{\mathrm{a}_k} - \left[\begin{array}{c}\boldsymbol{\varphi}_{1_k}^{\top} \\ \dots \\ \boldsymbol{\varphi}_{n_k}^{\top}\,\mathbf{A}_{n_k,1_k}\end{array}\right]\mathbf{A}_{1_k,\mathrm{b}}\,\mathbf{P}_{\mathrm{b}}\,\mathbf{X}_{\mathrm{b}}$$

which is slightly different from the fixed-base appendage case, where the RHS was only made up of $\bar{\mathbf{X}}_{\mathrm{a}_k}$. Developing the computation in the same fashion as in the third step of Appendix A.3.4, the i_k^{th} term of $\mathbf{X}_k$ satisfies the following property:

$$\mathbf{X}_{i_k} + \sum_{j_k=1_k}^{i_k-1}\mathbf{U}_{j_k,i_k}^{\top}\,\mathbf{X}_{j_k} = \bar{\mathbf{X}}_{\mathrm{a}_k} - \boldsymbol{\varphi}_{i_k}^{\top}\,\mathbf{A}_{i_k,1_k}\,\mathbf{A}_{1_k,\mathrm{b}}^{\top}\,\mathbf{P}_{\mathrm{b}}\,\mathbf{X}_{\mathrm{b}}$$

leading to:

$$\begin{aligned}\mathbf{X}_{i_k} &= \bar{\mathbf{X}}_{\mathrm{a}_k} - \boldsymbol{\varphi}_{i_k}^{\top}\,\mathbf{A}_{i_k,1_k}\,\mathbf{A}_{1_k,\mathrm{b}}^{\top}\,\mathbf{P}_{\mathrm{b}}\,\mathbf{X}_{\mathrm{b}} - \sum_{j_k=1_k}^{i_k-1}\boldsymbol{\varphi}_{i_k}^{\top}\,\mathbf{A}_{i_k,j_k}\,\mathbf{P}_{j_k}\,\mathbf{X}_{j_k} \\ &= \bar{\mathbf{X}}_{\mathrm{a}_k} - \boldsymbol{\varphi}_{i_k}^{\top}\,\mathbf{A}_{i_k,i_k-1}\underbrace{\left(\sum_{j_k=1_k}^{i_k-1}\mathbf{A}_{i_k-1,j_k}\,\mathbf{P}_{j_k}\,\mathbf{X}_{j_k} + \mathbf{A}_{i_k-1,1_k}\mathbf{A}_{1_k,\mathrm{b}}\,\mathbf{P}_{\mathrm{b}}\,\mathbf{X}_{\mathrm{b}}\right)}_{=\mu_{i_k-1}}\end{aligned}$$

where the variable μ_{i_k-1}, updated with the influence of the moving base, satisfies the same recursive relation:

$$\begin{aligned}
\mu_{i_k} &= \sum_{j_k=1_k}^{i_k} \mathbf{A}_{i_k,j_k} \mathbf{P}_{j_k} \mathbf{X}_{j_k} + \mathbf{A}_{i_k,\mathrm{b}}^{\top} \mathbf{P}_{\mathrm{b}} \mathbf{X}_{\mathrm{b}} \\
&= \mathbf{P}_{i_k} \mathbf{X}_{i_k} + \mathbf{A}_{i_k,i_k-1} \left(\sum_{j_k=1_k}^{i_k-1} \mathbf{A}_{i_k-1,j_k} \mathbf{P}_{1_k} \mathbf{X}_{1_k} + \mathbf{A}_{i_k-1,\mathrm{b}}^{\top} \mathbf{P}_{\mathrm{b}} \mathbf{X}_{\mathrm{b}} \right) \\
&= \mathbf{P}_{i_k} \mathbf{X}_{i_k} + \mathbf{A}_{i_k,i_k-1} \mu_{i_k-1}
\end{aligned}$$

The only change w.r.t. the fixed-base recurrence is the initial value of $\mu_{0_k} = \mathbf{A}_{0_k,\mathrm{b}}^{\top} \mathbf{P}_{\mathrm{b}} \mathbf{X}_{\mathrm{b}}$.

To summarize, the third step of the multi-module system is also solved by performing the third step for each module separately. It yields:

$$\mathbf{X}_{\mathrm{b}} = \bar{\mathbf{X}}_{\mathrm{b}} \tag{C.20a}$$

$$\forall k = 1 \dots n_{\mathrm{a}}, \quad \mathbf{X}_k = \mathbf{U}_{\mathrm{a}_k}^{-\top} \left(\bar{\mathbf{X}}_{\mathrm{a}_k} - \tilde{\mathbf{U}}_{\mathrm{a}_k}^{\top\ k} \mathbf{A}_{\mathrm{b}} \mathbf{P}_{\mathrm{b}} \mathbf{X}_{\mathrm{b}} \right) \tag{C.20b}$$

The whole forward dynamics algorithm is given in Algorithm C.3 based on the three recursions detailed in (C.17), (C.18) and (C.20).

Algorithm C.3: Forward dynamics algorithm for space robot

Function : $\ddot{\mathbf{q}} = \texttt{ForDynSpace}(\mathbf{b})$

Data : $\{\mathbf{A}_{i_k,i_k-1}\,,\,\mathbf{P}_{i_k}\,,\,\mathbf{M}_{i_k}\,,\,\mathbf{A}_{0_k,\mathrm{b}}\,,\,\mathbf{M}_\mathrm{b}\,,\,\mathbf{P}_\mathrm{b}\,;\; i_k = 0_k \dots n_k + 1\,,\, k = 1 \dots n_\mathrm{a}\}$

$\texttt{Initialize}\left(\boldsymbol{\eta}_\mathrm{b} = \mathbf{0}_{6\times1},\, \bar{\mathbf{M}}_\mathrm{b} = \mathbf{M}_\mathrm{b}\right)$ // **UPPER TRIANGULAR AND DIAGONAL SYSTEM**

1 **for** $k = 1 \dots n_\mathrm{a}$ **do**

Computation and storage of $\mathbf{A}_{i,i-1}$ (usually done in previous computation of $\mathbf{h}$ or $\mathbf{C}$) // Kinetics Loop

$\texttt{Initialize}\left(\boldsymbol{\eta}_{n_k+1} = \mathbf{0}_{6\times1},\, \bar{\mathbf{M}}_{n_k+1} = \mathbf{M}_{n_k+1},\, \bar{\boldsymbol{\varphi}}_{n_k+1} = \boldsymbol{\varphi}_{n_k+1} = \mathbf{0}_{6\times1}\right)$

2 **for** $i_k = n_k \dots 1_k$ **do**

3 $\boldsymbol{\eta}_{i_k,i_k+1} = \mathbf{A}^\top_{i_k+1,i_k}\,\boldsymbol{\eta}_{i_k+1}$ // Recursive computation of $\mathbf{U}_{\mathrm{a}_k}\widehat{\mathbf{X}}_{\mathrm{a}_k} = \mathbf{b}_k$

4 $\widehat{\mathbf{X}}_{i_k} = \mathbf{b}_{i_k} - \mathbf{P}^\top_{i_k}\,\boldsymbol{\eta}_{i_k,i_k+1}$ (A.41a)

5 $\bar{\mathbf{M}}_{i_k} = \mathbf{M}_{i_k} + \mathbf{A}^\top_{i_k+1,i_k}\left(\bar{\mathbf{M}}_{i_k+1} - \boldsymbol{\varphi}_{i_k+1}\,\bar{\boldsymbol{\varphi}}^\top_{i_k+1}\right)\mathbf{A}_{i_k+1,i_k}$ (A.34)

6 $\bar{\boldsymbol{\varphi}}_{i_k} = \bar{\mathbf{M}}_{i_k}\,\mathbf{P}_{i_k}$ (A.35)

7 $\boldsymbol{\Delta}_{i_k} = \mathbf{P}^\top_{i_k}\,\bar{\boldsymbol{\varphi}}_{i_k}$ (A.37)

8 $\boldsymbol{\varphi}_{i_k} = \bar{\boldsymbol{\varphi}}_{i_k}\boldsymbol{\Delta}^{-1}_{i_k}$ (A.36)

9 $\boldsymbol{\eta}_{i_k} = \boldsymbol{\varphi}_{i_k}\widehat{\mathbf{X}}_{i_k} + \boldsymbol{\eta}_{i_k,i_k+1}$ (A.41b)

// Recursive computation of $\boldsymbol{\Delta}_{\mathrm{a}_k}\bar{\mathbf{X}}_{\mathrm{a}_k} = \widehat{\mathbf{X}}_{\mathrm{a}_k}$

10 $\bar{\mathbf{X}}_{i_k} = \boldsymbol{\Delta}^{-1}_{i_k}\,\widehat{\mathbf{X}}_{i_k}$ (A.42)

11 $\mathbf{A}_{1_k,\mathrm{b}} = \mathbf{A}_{1_k,0_k}\,\mathbf{A}_{0_k,\mathrm{b}}$ // Intermediate variables of the spacecraft

12 $\boldsymbol{\eta}_\mathrm{b} = \boldsymbol{\eta}_\mathrm{b} + \mathbf{A}^\top_{1_k,\mathrm{b}}\boldsymbol{\eta}_{1_k}$

13 $\bar{\mathbf{M}}_\mathrm{b} = \bar{\mathbf{M}}_\mathrm{b} + \mathbf{A}^\top_{1_k,\mathrm{b}}\left(\bar{\mathbf{M}}_{1_k} - \boldsymbol{\varphi}_{1_k}\,\bar{\boldsymbol{\varphi}}^\top_{1_k}\right)\mathbf{A}_{1_k,\mathrm{b}}$

14 $\bar{\boldsymbol{\varphi}}_\mathrm{b} = \bar{\mathbf{M}}_\mathrm{b}\mathbf{P}_\mathrm{b}$ // Computation of $\mathbf{U}_\mathrm{b}\widehat{\mathbf{X}}_\mathrm{b} = \mathbf{b}_\mathrm{b}$ and $\boldsymbol{\Delta}_\mathrm{b}\bar{\mathbf{X}}_\mathrm{b} = \widehat{\mathbf{X}}_\mathrm{b}$

15 $\boldsymbol{\Delta}_\mathrm{b} = \mathbf{P}^T_\mathrm{b}\bar{\boldsymbol{\varphi}}_\mathrm{b}$

16 $\widehat{\mathbf{X}}_\mathrm{b} = \mathbf{b}_\mathrm{b} - \mathbf{P}^\top_\mathrm{b}\boldsymbol{\eta}_\mathrm{b}$

17 $\bar{\mathbf{X}}_\mathrm{b} = \boldsymbol{\Delta}^{-1}_\mathrm{b}\,\widehat{\mathbf{X}}_\mathrm{b}$

// **LOWER TRIANGULAR SYSTEM**

$\texttt{Initialize}\left(\mathbf{X}_\mathrm{b} = \bar{\mathbf{X}}_\mathrm{b},\, \boldsymbol{\mu}_\mathrm{b} = \mathbf{P}_\mathrm{b}\mathbf{X}_\mathrm{b}\right)$ // Computation of $\mathbf{U}^\top_\mathrm{b}\mathbf{X}_\mathrm{b} = \bar{\mathbf{X}}_\mathrm{b}$

18 **for** $k = 1 \dots n_\mathrm{a}$ **do**

19 $\boldsymbol{\mu}_{0_k} = \mathbf{A}_{0_k,\mathrm{b}}\boldsymbol{\mu}_\mathrm{b}$ // Computation of $\mathbf{U}^\top_{\mathrm{a}_k}\mathbf{X}_k = \bar{\mathbf{X}}_{\mathrm{a}_k}$

20 **for** $i_k = 1_k \dots n_k$ **do**

21 $\boldsymbol{\mu}_{i_k,i_k-1} = \mathbf{A}_{i_k,i_k-1}\boldsymbol{\mu}_{i_k-1}$

22 $\mathbf{X}_{i_k} = \bar{\mathbf{X}}_{i_k} - \boldsymbol{\varphi}^\top_{i_k}\,\boldsymbol{\mu}_{i_k,i_k-1}$ (A.43a)

23 $\boldsymbol{\mu}_{i_k} = \mathbf{P}_{i_k}\,\mathbf{X}_{i_k} + \boldsymbol{\mu}_{i_k,i_k-1}$ (A.43b)

---

24 **return** $\ddot{\mathbf{q}} = \mathbf{X}$

APPENDIX D MANIPULATOR AND SPACECRAFT DATA

This appendix gathers all the physical data used in simulation for the robotic arms and the spacecrafts. They are given in details in order to allow any simulation of the thesis to be reproduced faithfully.

D.1 Robotic Arms Data

The robotic arms data are provided for three different systems: the planar and the Canadarm examples used in the approximate model study in Section 5.2.6.1, and the manipulator designed by MDA for TAS. This latter is assumed to be embedded on the chaser spacecraft for the debris removal mission.

D.1.1 2DoF planar manipulator

The planar robot example of Fig. D.1 is taken from (De Luca and Siciliano, 1991). Its physical data are provided in Table D.1, along with the kinematic DH parameters for the three conventions. All masses are given in the International System of Units (SI): masses in kg, lengths in m, inertias in $kg.m^2$, and density in kg/m^3. Segments are considered homogeneous with the same material properties: $E = 1\,N/m^2$, $G = 1\,N/m^2$, $S = 1\,m^2$, $I_z = 1m^4$, $I_y = 1m^4$. Only one direction of bending is considered and two modes are used. As explained in (De Luca and Siciliano, 1991), a nominal configuration must be chosen to obtain the resulting loads of each segment, and then, to compute their clamped-loaded modes. The extended position is chosen here, with $\boldsymbol{\theta}_{nom} = \mathbf{0}_{2\times1}$.

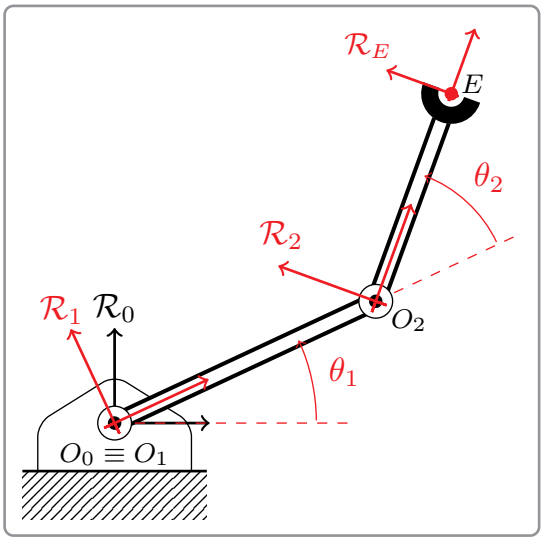


Figure D.1: Planar robot with 2 DoFs

With these parameters, the following pulsations are obtained for the two bending modes of each segment:

i	$\omega_{i,1}$	$\omega_{i,2}$
2	3.01	11.29
3	13.67	99.99

Reminding that mode shapes are normalized to ensure that $\mathbf{K}_{f_i}(j,j) = m_i\,\omega_{i,j}^2$ (De Luca and Siciliano, 1991), the corresponding stiffness matrices of each segment are obtained as :

$$\mathbf{K}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.91 & 0 \\ 0 & 0 & 12.74 \end{bmatrix} \qquad \mathbf{K}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 18.73 & 0 \\ 0 & 0 & 999.88 \end{bmatrix}$$

These results were computed using Appendix B.1, and prove to be consistent with the data provided in (De Luca and Siciliano, 1991). It is important to recall that the modes were assumed to be orthogonal, and that the translational and rotary inertias were given by:

$$\mathbf{M}_{\delta\delta,1} = m_1\,\mathbf{I}_2 \qquad \mathbf{M}_{\delta\delta,2} = m_2\,\mathbf{I}_2$$

If these inertia are computed with the AMM, they result in overestimated rotary inertias because the material properties above do not have any physical meaning, and are not correlated with the inertia tensor of the segments. So these later are neglected in the simulation of this planar example.

Table D.1: Data of 2 DoFs planar robot (SI units)

Dynamic parameters

i	ρ_i	l_i	m_i	J_i^z
1	0.2	0.5	0.1	0.0083
2	0.2	0.5	0.1	0.0083

Hub parameters

i	$m_{h_i}[kg]$	$J_{h_i}^z[kg.m^2]$
1	1	0.1
2	1	0.1
Payload	0.1	0.0005

Classic DH parameters

i	θ_i	d_i	a_i	α_i
1	θ_1	0	l_1	0
2	θ_2	0	l_2	0
E	0	0	0	0

Modified DH parameters

i	α_i	a_i	d_i	θ_i
1	0	0	0	θ_1
2	0	l_1	0	θ_2
E	0	l_2	0	0

Adapted DH parameters

i	d_i	a_i	α_i	θ_i
1	0	0	0	θ_1
2	0	l_1	0	θ_2
E	0	l_2	0	0

Flexible data (bending only)

i	$\phi_{i,1e}$	$\phi_{i,2e}$	$\phi'_{i,1e}$	$\phi'_{i,2e}$	$v_{i,1}$	$v_{i,2}$
1	0.1859	0.2151	0.6571	−0.5604	0.0066	0.0131
2	0.8833	−0.0693	2.6413	−10.8526	0.0333	0.0544

i	$w_{i,1}$	$w_{i,2}$	$z_{i,11}$	$z_{i,12}$	$z_{i,22}$
1	0.0024	0.0045	0.0007	0.0013	0.0024
2	0.0122	0.0156	0.0185	0.0205	0.0406

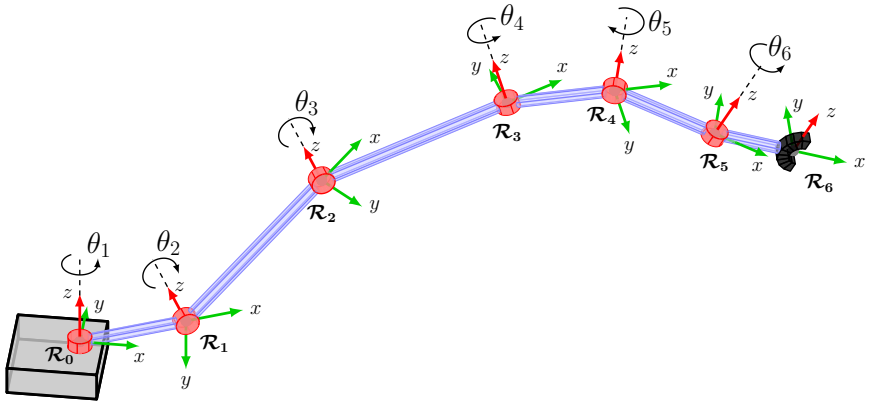


Figure D.2: Canadarm manipulator with 6 DoFs

D.1.2 6DoF Canadarm manipulator

The second example is the *Space Shuttle Remote Manipulator System*, also called Canadarm. Its kinematic scheme with the DH frames is given in Figure D.2 and its corresponding data are gathered in Table D.2. They are taken from (Mohan and Saha, 2009). At the flexible level, only the second and third segments are considered flexible. Ten modes are used for both segments, with one in traction, four in each bending direction and one in torsion. The fully extended position is still chosen to compute the loads for these clamped-loaded modes: $\boldsymbol{\theta}_{nom} = \mathbf{0}_{6\times1}$. The load mass and inertia are therefore identical for each segment along the Y and Z axes[1] since the manipulator exhibit a symmetry of revolution in this configuration. As a consequence, the same bending modes are obtained in both Y and Z directions. Therefore, flexible modes data given in Table D.2 are given for Y-bending only, since the same values are used for Z-bending. Another nominal configuration with asymmetric loads would lead to different modes. These modes are still normalized by the stiffness matrix, i.e., $\mathbf{K}_{f_i}(j,j) = m_i\,\omega_{i,j}^2$.

Segments are considered tubular of radius $7.5\,cm$ and thickness $2\,cm$ ($S = 0.01\,m^2$), with the following material properties:

$$E = 2.5\times10^9\,N/m^2 \qquad I_z = 3.91\times10^{-5}\,m^4$$
$$G = 7.17\times10^9\,N/m^2 \qquad I_y = 3.91\times10^{-5}\,m^4$$

With these parameters, the following pulsations are obtained for the flexible modes of second and third segments (expressed in *rad/s*):

i	$\omega_{i,x,1}$	$\omega_{i,y,1}$	$\omega_{i,y,2}$	$\omega_{i,y,3}$	$\omega_{i,y,4}$	$\omega_{i,\alpha,1}$
2	127.03	1.07	4.83	42.17	112.98	215.12
3	149.01	2.16	11.22	44.35	115.55	252.35

[1] of the beam frame, set with the X-axis aligned with the beam axis.

Table D.2: Data of the 6 DoFs Canadarm manipulator (SI units)

Dynamic parameters

i	ρ_i	l_i	m_i	J_i^x	J_i^y	J_i^z
1	4447	1	47.5	0.20	38.12	38.12
2	2185	6	140	0.40	639.99	639.99
3	1137	7	85	0.20	358.77	358.77
4	4447	1	47.5	0.20	38.12	38.12
5	4447	1	47.5	0.20	38.12	38.12
6	4447	1	47.5	0.20	38.12	38.12

Classic DH parameters

i	θ_i	d_i	a_i	α_i
1	θ_1	0	l_1	$-\pi/2$
2	θ_2	0	l_2	0
3	θ_3	0	l_3	$\pi/2$
4	θ_4	0	l_4	$-\pi/2$
5	θ_5	0	l_5	$\pi/2$
6	θ_6	0	l_6	0
E	0	0	0	0

Modified DH parameters

i	α_i	a_i	d_i	θ_i
1	0	0	0	θ_1
2	$-\pi/2$	l_1	0	θ_2
3	0	l_2	0	θ_3
4	$\pi/2$	l_3	0	θ_4
5	$-\pi/2$	l_4	0	θ_5
6	$\pi/2$	l_5	0	θ_6
E	0	l_6	0	0

Adapted DH parameters

i	d_i	a_i	α_i	θ_i
1	0	0	0	θ_1
2	0	l_1	$-\pi/2$	θ_2
3	0	l_2	0	θ_3
4	0	l_3	$\pi/2$	θ_4
5	0	l_4	$-\pi/2$	θ_5
6	0	l_5	$\pi/2$	θ_6
E	0	l_6	0	0

Flexible data

i	$\phi_{i,x,1e}$	$\phi_{i,y,1e}$	$\phi_{i,y,2e}$	$\phi_{i,y,3e}$	$\phi_{i,y,4e}$	–
2	0.7102	0.3135	0.6338	−0.2181	0.1432	–
3	0.7013	0.6814	0.1675	−0.2014	0.1382	–
i	–	$\phi'_{i,y,1e}$	$\phi'_{i,y,2e}$	$\phi'_{i,y,3e}$	$\phi'_{i,y,4e}$	$\phi_{i,\alpha,1e}$
2	–	0.0987	−0.0458	0.0060	−0.0023	8.2980
3	–	0.1501	−0.3989	0.1182	−0.0444	8.1936
i	$v^u_{i,x,1}$	$v^u_{i,y,1}$	$v^u_{i,y,2}$	$v^u_{i,y,3}$	$v^u_{i,y,4}$	–
2	51.93	15.05	48.06	106.02	2.62	–
3	31.10	21.55	28.74	61.55	1.56	–
i	–	$v^p_{i,y,1}$	$v^p_{i,y,2}$	$v^p_{i,y,3}$	$v^p_{i,y,4}$	$v^p_{i,\alpha,1}$
2	–	684.92	1384.54	−476.44	312.81	56804.5
3	–	774.59	190.47	−228.94	157.07	34014.7
i	$w^u_{i,x,1}$	$w^u_{i,y,1}$	$w^u_{i,y,2}$	$w^u_{i,y,3}$	$w^u_{i,y,4}$	–
2	205.93	67.34	199.41	297.91	−142.25	–
3	143.90	110.78	124.55	197.69	−100.09	–

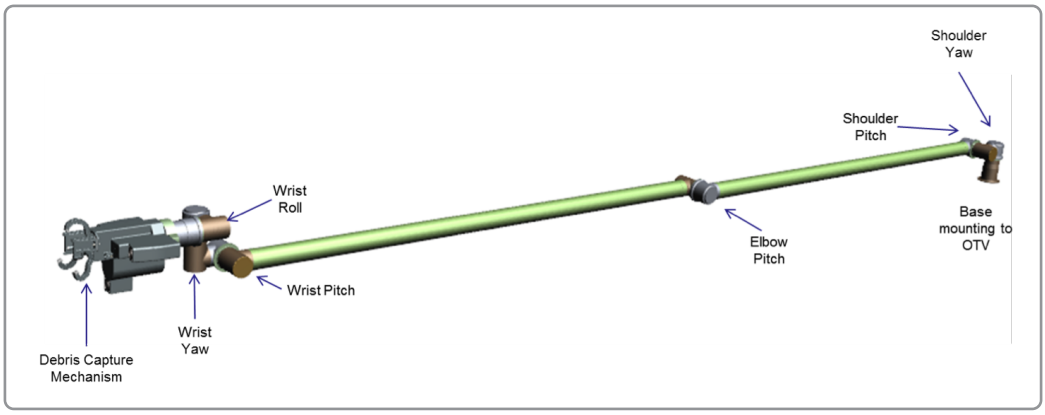


Figure D.3: Robotic manipulator with 6 DoFs designed by MDA

D.1.3 6DoF manipulator of the space mission

The robotic arm used during the simulations was designed by the Canadian company *MacDonald, Dettwiler and Associates (MDA)* and is illustrated in Figure D.3. This manipulator possesses 6 joints and has a structure similar to the Canadarm presented above. The corresponding DH parameters and the dynamic and flexible data are summarized in Table D.3.

Similarly to the previous example, only the second and third segments are considered flexible. Nevertheless, the modes have different modal participation matrices for this structure, because the load of each segment is completely asymmetric. As a result of previous simulations, the highest flexible modes have a small influence on the simulations, so they were reduced in bending for this example. Six modes are used to describe each of the two segments: one in traction, two in both bending directions, and a last one in torsion. The load are also computed on the nominal configuration of an arm fully extended, with $\boldsymbol{\theta}_{nom} = \mathbf{0}_{6\times1}$. Concerning the material properties, they are inspired by the Canadarm ones, available in the literature or on the CSA website. They are given as follows for tubular segments of diameter 10 cm, thickness of 4.8 mm and section surface of 1.6 mm^2:

$$E = 10.3 \times 10^9\, N/m^2 \qquad I_z = 2.15 \times 10^{-6}\, m^4$$
$$G = 7.17 \times 10^9\, N/m^2 \qquad I_y = 2.15 \times 10^{-6}\, m^4$$

In addition, the hubs were modeled using data provided by MOOG motors[2]. The motor inertias were set to $5.51 \times 10^{-5}\, kg.m^2$, their gear ratio to 450, and their mass to $5.4\, kg$. A payload is also considered to model the tool with a mass of $15.8\, kg$ at the end-tip of the last segment. The resulting flexible frequencies are given as follows (in *rad/s*):

i	$\omega_{\mathbf{i,x,1}}$	$\omega_{\mathbf{i,y,1}}$	$\omega_{\mathbf{i,y,2}}$	$\omega_{\mathbf{i,z,1}}$	$\omega_{\mathbf{i,z,2}}$	$\omega_{\mathbf{i},\alpha,\mathbf{1}}$
2	344.43	4.65	19.34	4.75	19.47	27.37
3	405.39	9.59	53.99	9.01	38.73	27.19

[2]http://www.moog.com/products/motors-servomotors/brushless-motors/high-performance-direct-drive-brushless-dc-motors/db-4530-a-1es/

Table D.3: Data of the 6 DoFs manipulator designed by MDA (SI units)

Dynamic parameters

i	ρ_i	l_i	m_i	J_i^x	J_i^y	J_i^z
1	2250	0.10	0.36	0.82	1.00	0.82
2	–	2.61	9.20	0.03	5.24	5.24
3	–	2.61	9.20	0.03	5.24	5.24
4	–	0.10	0.36	1.00	0.82	0.82
5	–	0.10	0.36	0.82	0.82	1.00
6	–	0.50	1.78	0.04	0.04	0.01

Classic DH parameters

i	θ_i	d_i	a_i	α_i
1	θ_1	0.06	l_1	$\pi/2$
2	θ_2	0.10	l_2	0
3	θ_3	0.10	l_3	0
4	θ_4	0.10	l_4	$-\pi/2$
5	$\theta_5+\pi/2$	0.14	l_5	$\pi/2$
6	θ_6	l_6	0	0
E	0	0	0	0

Modified DH parameters

i	α_i	a_i	d_i	θ_i
1	0	0	0.06	θ_1
2	$\pi/2$	l_1	0.10	θ_2
3	0	l_2	0.10	θ_3
4	0	l_3	0.10	θ_4
5	$-\pi/2$	l_4	0.14	$\theta_5+\frac{\pi}{2}$
6	$\pi/2$	l_5	l_6	θ_6
E	0	0	0	0

Adapted DH parameters

i	d_i	a_i	α_i	θ_i
1	0	0	0	θ_1
2	0.06	l_1	$\pi/2$	θ_2
3	0.10	l_2	0	θ_3
4	0.10	l_3	0	θ_4
5	0.10	l_4	$-\pi/2$	$\theta_5+\pi/2$
6	0.14	l_5	$\pi/2$	θ_6
E	l_6	0	0	0

Flexible data

i	$\phi_{i,x,1e}$	$\phi_{i,y,1e}$	$\phi_{i,y,2e}$	$\phi_{i,z,1e}$	$\phi_{i,z,2e}$	–
2	0.9695	−1.2183	−2.0488	−1.2129	−1.9799	–
3	0.9570	−0.9194	−0.2577	−0.8594	−0.4145	–

i	–	$\phi'_{i,y,1e}$	$\phi'_{i,y,2e}$	$\phi'_{i,z,1e}$	$\phi'_{i,z,2e}$	$\phi_{i,\alpha,1e}$
2	–	−0.8638	0.5019	0.8572	−0.5129	0.9997
3	–	−0.5437	1.5674	0.5260	−0.9873	0.9997

i	$v^u_{i,x,1}$	$v^u_{i,y,1}$	$v^u_{i,y,2}$	$v^u_{i,z,1}$	$v^u_{i,z,2}$	–
2	4.5263	−3.8765	−10.4703	−3.8649	−10.1746	–
3	4.4942	−3.1442	−4.4847	−2.9031	−3.9523	–

i	–	$v^p_{i,y,1}$	$v^p_{i,y,2}$	$v^p_{i,z,1}$	$v^p_{i,z,2}$	$v^p_{i,\alpha,1}$
2	–	−2741.2	−4609.9	2729.0	4454.7	2935.9
3	–	−2068.7	−579.8	1933.7	932.7	2935.9

i	$w^u_{i,x,1}$	$w^u_{i,y,1}$	$w^u_{i,y,2}$	$w^u_{i,z,1}$	$w^u_{i,z,2}$	–
2	7.8525	−7.5332	−18.8474	−7.5085	−18.3018	–
3	7.7878	−6.0271	−7.2953	−5.5776	−6.6713	–

D.2 Spacecraft Data

The data used for the spacecraft are provided below. The models used in simulation for the chaser and for the debris are based on real data available online or in the literature.

D.2.1 Chaser Spacecraft

Base Concerning the spacecraft of the chaser, the data are taken from the DEOS project, based on the website of the company SpaceTech GmbH[3] and of the DLR. They are summarized as follows in SI units, assuming an homogeneous cubic shape:

Height	Width	Depth	Mass	J_x	J_y	J_z
2.6	1.7	1.8	786	401.5	655.0	632.1

Solar Array In addition, the solar panels of the spacecraft are also modeled as flexible elements. Since the behavior of plate members is very complex, only a linear model is proposed here. The rigid parameters of the solar arrays are defined for a cubic shape by:

Height	Width	Depth	Mass	J_x	J_y	J_z
4	2	0.05	200	66.7	333.3	266.7

Their flexible parameters are given as follows for $n_{\text{sa}} = 3$ modes in both bending directions, and in torsion:

	y – Bending	z – Bending	Torsion
ω_i	8	15	20
ξ_i	0.005	0.005	0.005

The mode frequencies and damping allow to define the flexible mass and stiffness matrices:

$$\mathbf{M}_{\text{f}} = \begin{bmatrix} \mathbf{M}_{\text{r}} & \mathbf{L}^{\top} \\ \mathbf{L} & \mathbf{I}_{n_{\text{sa}}} \end{bmatrix} \qquad \mathbf{K}_{\text{f}} = \text{diag}(\mathbf{0}_{6\times 6}, \omega_i^2) = \left[\begin{array}{c|ccc} \mathbf{0}_{6\times 6} & 0 & 0 & 0 \\ \hline 0 & 64 & 0 & 0 \\ 0 & 0 & 225 & 0 \\ 0 & 0 & 0 & 400 \end{array}\right]$$

[3] http://www.spacetech-i.com/images/press/downloads/DEOS_PhaseA_Datasheet.pdf

with $\mathbf{M}_{\mathrm{r}}$ the rigid mass matrix of the solar array, and $\mathbf{L}$ its modal participation, which is alternatively defined by $\mathbf{P}_{ru}$ for the translation, and $\mathbf{H}_{ru}$ for the rotation in the mass matrix expression of a segment in (B.46). Its explicit content reads as follows:

$$\mathbf{L} = (100^{1/2}) \begin{bmatrix} 0 & 0.5 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0.2 & 0 & -0.2 & 0 \\ 0 & 0 & 0 & 0.2 & 0 & 0 \end{bmatrix}$$

D.2.2 Target Spacecraft

The debris corresponds to a real satellite whose data are provided by TAS. The shape is also considered cubic, as illustrated in Figure D.4. However, the mass distribution is taken into account through non-zero inertia cross terms. The dynamic data of the debris are summarized as follows in SI units:

Height	Width	Depth	Mass	J_x	J_y	J_z	J_{xy}	J_{yz}	J_{xz}
3.77	1.50	1.50	1060	1440	1120	2250	−64	18	24

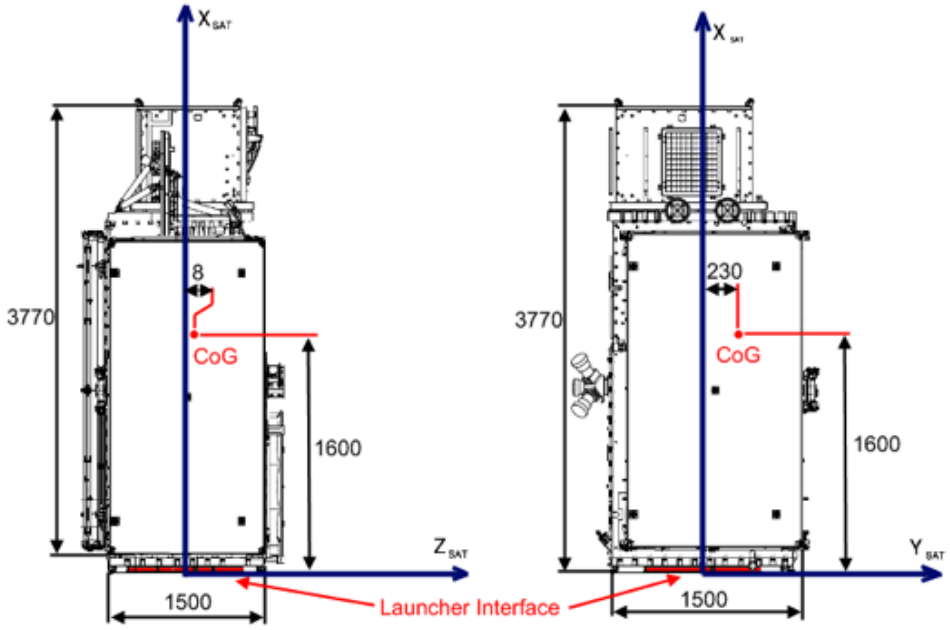


Figure D.4: Dimensions of the debris provided by TAS